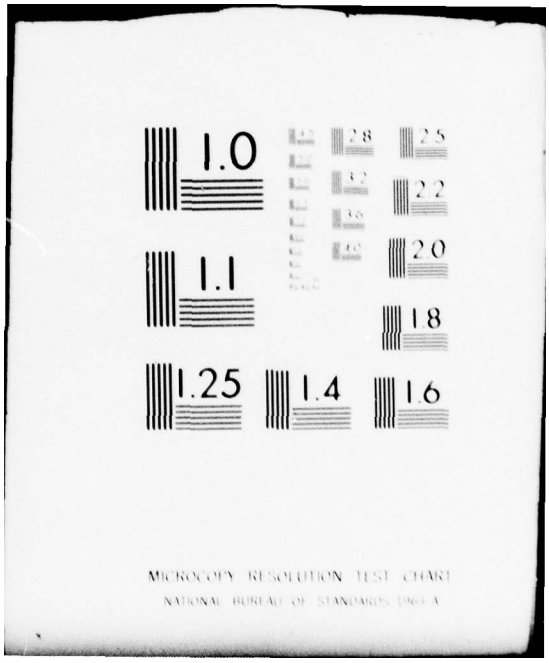
AD-A068 310 AERONAUTICAL SYSTEMS DIV WRIGHT-PATTERSON AFB OH F/G 17/2
PROCEEDINGS OF THE AFSC MULTIPLEX DATA BUS CONFERENCE (2ND), HE--E
OCT 78 E C GANGL
UNCLASSIFIED ASD-TR-78-34-VOL-2 NL

1 OF 9
AD A068310




1.0
1.1
1.25
1.4
1.6
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.8
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

AD A068310

DDC FILE COPY

ASD-TR-78-34

# 2nd AFSC MULTIPLEX DATA BUS CONFERENCE

LEVEL II

DDC RECEIVED MAY 7 1979 C

10 - 12 OCTOBER 1978

DAYTON CONVENTION CENTER

DAYTON, OHIO

SPONSORED BY:

AIR FORCE SYSTEMS COMMAND

## PROCEEDINGS

FINAL

APPROVED FOR PUBLIC RELEASE;
DISTRIBUTION UNLIMITED

HOSTED BY:

AERONAUTICAL SYSTEMS DIVISION

79 04 05 021

## NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This report has been reviewed by the Information Office (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

ERWIN C. GANGL
Technical Advisor
Information Engineering Division

LOUIS J. URBAN
Technical Director, Avionics Engineering
Deputy for Engineering

HAROLD C. PLUENNEKE, Colonel, USAF
Deputy for Engineering
Aeronautical Systems Division

GEORGE L. VON TERSCH, Colonel, USAF
Director, Avionics Engineering
Deputy for Engineering

"If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify ASD/ENAI, W-PAFB, OH 45433 to help us maintain a current mailing list".

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

AIR FORCE/56780/6 April 1979 — 1000

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

| REPORT DOCUMENTATION PAGE | READ INSTRUCTIONS BEFORE COMPLETING FORM |
|---|---|
| 1. REPORT NUMBER: ASD-TR-78-34 Vol-2 (14) | 2. GOVT ACCESSION NO. / 3. RECIPIENT'S CATALOG NUMBER |
| 4. TITLE (and Subtitle): Proceedings of the AFSC Second Multiplex Data Bus Conference (2nd), held at Dayton Convention Center, Dayton, Ohio, 10-12 October 1978. Volume II. (6) | 5. TYPE OF REPORT & PERIOD COVERED: Vol II of II, 10-12 October 1978 |
| | 6. PERFORMING ORG. REPORT NUMBER |
| 7. AUTHOR(s): Editor: Erwin C. Gangl (10) | 8. CONTRACT OR GRANT NUMBER(s): Final rept. (9) |
| 9. PERFORMING ORGANIZATION NAME AND ADDRESS: HQ ASD/ENAI, Wright-Patterson AFB, Ohio 45433 | 10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS (11) (12) 851p. |
| 11. CONTROLLING OFFICE NAME AND ADDRESS: HQ ASD/ENAI, Wright-Patterson AFB, Ohio 45433 | 12. REPORT DATE: 10 - 12 October 1978 |
| | 13. NUMBER OF PAGES: 886 |
| 14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): Same as above. | 15. SECURITY CLASS. (of this report): UNCLASSIFIED |
| | 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE |

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

N/A

18. SUPPLEMENTARY NOTES

N/A

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Multiplexing, MIL-STD-1553, Data Bus, Fibre Optics, Digital Avionics, Digital Integration, Systems Integration, Systems Architecture, Standardization, Information Transfer.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This is a collection of unclassified papers presented by U.S. and international government and industry personnel at the second AFSC Multiplex Data Bus Conference at the Convention Center, Dayton, Ohio. The purpose of the conference is to exchange information on data bus technology and collect lessons learned for future systems applications. Detailed abstracts cannot be provided here, but can be found at the beginning of each paper.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

008 800

ASD-TR-78-34
Vol II of II

# PROCEEDINGS, FINAL Copy *

of the

# 2ND AFSC MULTIPLEX DATA BUS CONFERENCE

10-12 October 1978

Dayton Convention Center

Dayton, Ohio

Approved for public release;
distribution unlimited.

Sponsored by: AIR FORCE SYSTEMS COMMAND

Hosted by: AERONAUTICAL SYSTEMS DIVISION

Organized by: DEPUTY FOR ENGINEERING (ASD/ENAI)
AERONAUTICAL SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433

*NOTE: This FINAL Copy supersedes Vol. I of ASD-TR-78-34 (NH)

WELCOME
COLONEL WALKER A. LARIMER
DEPUTY FOR AVIONICS CONTROL
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433

KEYNOTE ADDRESS
COLONEL ROBERT S. ZIERNICKI
ASSISTANT DEPUTY CHIEF OF STAFF
DEVELOPMENT PLANS FOR INTEGRATION AND ARCHITECTURE
AIR FORCE SYSTEMS COMMAND
ANDREWS AIR FORCE BASE MD 20334

## FOREWORD

The first Multiplex Data Bus Conference was held in Dayton, Ohio 3-5 November 1976. Its Proceedings, ASD-TR-76-22, ADA-033793, has been used the past two years as a primary source of information for data bus designers. Significant advances in systems applications, LSI developments and the preparation of MIL-STD-1553B have prompted this second Multiplex Data Bus Conference. Maj Gen Toomay, Deputy Chief of Staff, Development Plans, Air Force Systems Command, requested this conference being hosted by the Aeronautical Systems Division, Lt General Sylvester, Commander.

The purpose of the conference is to exchange data on lessons learned from system applications, present new LSI hardware developments, exploit new data bus technology and present test methods for MIL-STD-1553 multiplexed data buses between the U.S. Air Force, Navy, Army, NASA, industry and foreign governments/industry.

This is the FINAL Volume, Volume II of the 2nd MUX Conference. Volume I had only those papers that reached us in time for printing prior to the conference. To provide a better reference document, this FINAL Volume has reprinted all the papers from Volume I and also reprinted MIL-STD-1553B with the typographical errors removed.

Many thanks to Major Robert W. Johnson (AFSC/XRF) for his headquarters' assistance in organizing this conference and also to the ASD/ENAI staff who assisted in organizing the technical program, proceedings and exhibits, Captain Frederick Pensworth and Mr. Harold Alber. Special thanks to Mr. Allen J. Cannon (ASD/CS) who did an outstanding job in organizing the conference facilities, hotel accommodations and food service.

Thanks also to the moderators and all the speakers who responded with outstanding presentations and papers in a timely manner despite such short notice. And, finally to my secretary, Mrs. Marie Jankovich, for her expert administrative help in handling the conference correspondence and registration.

Erwin C. Gangl

ERWIN C. GANGL, Technical Advisor
ASD/ENAI
Conference Organizer

ACCESSION for
NTIS White Section
DDC Buff Section
UNANNOUNCED
JUSTIFICATION
BY
DISTRIBUTION/AVAILABILITY CODES
Dist SPECIAL
A

DEPARTMENT OF THE AIR FORCE
HEADQUARTERS AIR FORCE SYSTEMS COMMAND
ANDREWS AIR FORCE BASE, DC 20334

REPLY TO ATTN OF: XR

MAY 15 1978

SUBJECT: Multiplex Data Bus Conference

TO: ASD/CV

1. The 1976 Conference on Multiplex Data Buses that you hosted was highly successful. Since there has been significant technological progress in this area and widespread international interest in the U.S. Military Standard Multiplex Bus, MIL-STD-1553, request you host a follow-on conference in 1978.

2. In your 17 December 1976 letter to AFSC, "Multiplex Data Bus Conference Recommendations," you pointed out that life-cycle savings and interoperability can only result by exercising control over, and sharing lessons learned on multiplexed data buses. This follow-on conference should, therefore, stress our experiences from system applications, the handbook that was developed using the information collected during the 1976 Conference, and the current large scale integration (LSI) circuit developments of multiplex data bus components. The on-going work to control MIL-STD-1553 and provide an upward compatible update to it based on these experiences should also be presented.

3. The AFSC project officer for this effort, Maj Robert W. Johnson (AFSC/XRF), AUTOVON 858-5731, is prepared to provide you with any assistance you might need in arranging this conference.

FOR THE COMMANDER

[illegible] C. [illegible], Maj Gen, USAF
DCS/Development Plans

PROCEEDINGS

MULTIPLEX DATA BUS CONFERENCE

10-12 OCTOBER 1978

DAYTON, OHIO


| | Page |
|---|---|
| SESSION I: "CURRENT DATA BUS APPLICATIONS"<br>Moderator: John R. Cannon, ASD/XRE, W-PAFB, OH | 1 |
| F-16 MULTIPLEX: A SYSTEMS PRESPECTIVE<br>D. E. Sundstrom, W. B. Anderson, Jr., and S. A. Alford, General Dynamics Ft. Worth Division, Ft. Worth, TX | 3 |
| THE APPLICATION OF MIL-STD-1553A TO THE LAMPS MK-III AIRBORNE SYSTEM, Fred Shocket and Ralph E. Hungerford, Naval Air Development Center, Warminster, PA | 29 |
| FRENCH UTILIZATION OF THE DIGITAL MULTIPLEX BUS<br>Bernard Vandecasteele, Ministry of Defense, Service Technique Aeronautique, France | 45 |
| MULTIPLEXING IN ARMY TANK-AUTOMOTIVE VEHICLES<br>Jerry L. Perry, Systems Consultants, Inc. and Marquis W. Woody, U.S. Army Tank-Automotive, R&D Command, Warren, MI | 51 |
| MINISTRY OF DEFENSE PROGRAMS IN SUPFORT OF MIL-STD-1553 RELATED RESEARCH, Dr. A. A. Callaway, Royal Aircraft Establishment, United Kingdom | 61 |
| SOFTWARE APPROACH TO MULTICOMPUTER SYSTEM ARCHITECTURE USING MIL-STD-1553 DATA BUS<br>H. Niedzballa and H. Ernst, ESG Elektronik-System, Germany | 63 |
| APPLICATIONS OF ASYNCHRONOUS MULTIPLEX DATA BUS IN THE TRIDENT TRAINING SITE, James Baker, General Electric | 93 |
| SESSION II: "SYSTEMS INTEGRATION USING MIL-STD-1553"<br>Moderator: Colonel Eric B. Nelson, ESD/DCB, Hanscom AFB, MA | 95 |
| H-X NIGHT/ADVERSE-WEATHER AVIONICS STUDY<br>Cheryl L. Zelasco, ASD/ENAS, W-PAFB, OH | 97 |
| MULTIPLEX SYSTEM FOR THE HUGHES ADVANCED ATTACK HELICOPTER - YAH-64, Frank Booth, Hughes Helicopters, Culver City, CA and David Espen, Sperry Flight Systems, Phoenix, AZ | 144 |

Page

EVOLUTION OF A MULTIPLEX SYSTEM 165
George Cotter, Larry Driscoll, Howard Sherman,
Grumman Aerospace Corp., Bethpage, NY

INVESTIGATION OF DATA SYSTEMS ARCHITECTURES FOR STORES 183
MANAGEMENT, Dr. William A. Hornfeck, Gannon College, Erie, PA,
and John Cavin and Ron Boulet, AF Armament Lab, Eglin AFB, FL

MULTIPLEXING IN AIDS 195
John F. Moran, General Electric Co., Utica, NY

MULTIPLEX BUS FOR INTEGRATED AVIONICS MANAGEMENT SYSTEMS 205
Merrill T. Ludvigson, Rockwell-International, Cedar Rapids, IA

PRECISION ATTACK ENHANCEMENT INTEGRATION 221
Capt Ron Marini, ASD/AER, W-PAFB, OH

MULTIPLEX CONTROL OF AIRCRAFT SUBSYSTEMS 223
F. E. Funke, Rockwell International, El Segundo, CA

SESSION III: "NEW DATA BUS CONCEPTS" - PART I 253
Moderator: Lt Col John Marciniak, AFSC/XRF, Andrews AFB, MD

ULAIDS - 1553 ARCHITECTURE WITH DYNAMIC BUS ALLOCATION 255
Louis Ciasulli & Patrick Henderson, Conrac Corp, W. Caldwell, NJ

GINA AIRBORNE DIGITAL DATA BUS, A. Quidet, Electronique 321
Marcel Dassault, France

1553 EXTENSIONS WHICH ALLOW ELIMINATION OF CENTRAL CONTROLLERS 351
H. Kister, VDO, Germany

USE OF DIGITAL DATA BUSES ON COMMERCIAL AIRCRAFT 381
V. J. Small, Boeing Commercial Airplane Co.

ARINC 575-TO-1553A INTERFACE 391
John F. Howard, ASD/ENAIB, W-PAFB, OH

BUS CONTROLLER SOFTWARE IMPLEMENTATION 401
Thomas L. Zawodny, ASD/ENAIA (SEAFAC), W-PAFB, OH

AN AIRCRAFT COMMUNICATIONS MANAGEMENT TERMINAL (CMT) 409
T. Gedeon and S. Thornton, SCI, Huntsville, AL

SESSION IV: "NEW DATA BUS CONCEPTS" - PART II 411
Moderator: Dr. William E. Eppers, Jr., AFAL, W-PAFB, OH

MULTIPLEX CONSIDERATIONS AND POTENTIAL APPLICATIONS IN 413
MODULAR AUTOMATIC TEST EQUIPMENT (MATE)
John E. Scott, Sperry Rand Corp., Great Neck, NY

DISTRIBUTED PROCESSING FOR MIL-STD-1553A 423
Carl Wise, Westinghouse, Baltimore, MD and
James Bain, AFAL, W-PAFB, OH

Page

INTERFACES AND DATA TRANSMISSION ON A PARALLEL BUS FOR MODULAR EQUIPMENT, Dr. Walter Metzdorff and Dr. Manfred Mall, Dornier Systems GMBH, Germany 431

GENERAL PURPOSE AUDIO-MULTIPLEXED INTERCOM SYSTEM
AiResearch Mfg Co. of California, Torrance, CA 443

DATA BUS SYSTEMS FOR DISCRETE SIGNALS
H. Kister, VDO, Germany 459

DEVELOPMENT OF THE MIL-STD-1553 MULTIPLEX APPLICATIONS HANDBOOK, C. Ray Turner, The Boeing Co , Seattle, WA 489

A COMPARISON OF TIME DIVISION MULTIPLEX DATA BUS TECHNIQUES BASED ON DATA TRANSFER PERFORMANCE
Charles R. Husbands, The MITRE Corp., Bedord, MA 523

SESSION V: "DATA BUS HARDWARE" 533
Moderator: Colonel Clifford H. Allen, AFATL/DLM, Eglin AFB, FL

USE OF STANDARD MODULES FOR AIRBORNE MULTIPLEXING APPLICATIONS
J. W. McCuen & G. E. Smith, Hughes A/C Co., Newport Beach, CA 535

DUAL CHANNEL 1553A INTERFACE
Robert Hoffman, Applied Technology/ITEK Corp., Sunnyvale, CA 547

A FLEXIBLE MULTIPLEX TERMINAL INTERFACE UNIT
Remigius Shatas & Jerry Blevins, SCI Systems Inc., Huntsville, AL 559

THE DESIGN AND DEVELOPMENT OF AN LSI BUS INTERFACE UNIT
Andrew Gutierrez & Maj Gary Pritchard, AFAL, W-PAFB, OH 581

THE BUS INTERFACE UNIT - AN ADAPTABLE LSI CHIP-SET
Daniel G. Yelverton, Harris Corp., Melbourne, FL 589

SOME APPLICATIONS OF THE ROCKWELL DATA TERMINAL CHIP
B. E. Bona, Autonetics Strategic Systems, Anaheim, CA 609

A HYBRIDIZED 1553A INTERFACE
W. R. Hunter, Teledyne Systems Co., Northridge, CA 625

TOWARDS A UNIVERSAL SYNCHRONOUS COMMUNICATIONS CHANNEL UTILIZING MICROPROGRAMMED DEVICES
L. L. King, Westinghouse, Baltimore, MD 641

SESSION VI: "TESTING AND EVALUATION OF MULTIPLEX SYSTEMS" 651
Moderator: Colonel George L. Von Tersch, ASD/ENA, W-PAFB, OH

1553 BUS TESTERS
Nancy L. Esken/Anthony L. Haley, ASD/ENASC, W-PAFB, OH 653

Page

MIL-STD-1553 MONITOR INTERFACE FOR FLIGHT TEST 657
James L. Voight & Donald H. Ellis, SCI Systems, Huntsville, AL

LOGIC STRUCTURE FOR TESTABILITY AND FAILURE DETECTION 671
U. Schulz and A. Roelker, Dornier System GMBH, Germany

DIGITAL DATA ACQUISITION SYSTEM (D-DAS)-A FLEXIBLE TOOL FOR SIMULATION, DEBUG, AND TRAFFIC MONITORING OF 1553 SYSTEMS 683
Michael Boice, EMM/SESCO

DATA BUS SIMULATION AND TESTING ON THE SPACE SHUTTLE PROGRAM 691
Michael R. Sottile, Singer-Kearfott, Little Falls, NJ

AIRCRAFT MULTIPLEX SYSTEM DESIGN AND INTEGRATION REQUIREMENTS FOR LOW NOISE DATA BUS OPERATION 719
William S. Hermann & Dr. Clifford D. Skouby, McDonnell A/C Co., St. Louis, MO

MULTIPLEX DATA BUS APPLICATIONS IN AIRCRAFT SIMULATORS 735
Chris G. Horattas, Goodyear Aerospace Corp., Akron, OH

1553B: HOW DO YOU KNOW YOU WILL PASS THE NOISE TEST? 749
Robert Salter, Sperry Univac

SPECIAL SESSION: "APPLICATIONS OF FIBER OPTICS TO MULTIPLEX SYSTEMS" 761
Moderator: Francis Lueking, Air-360, NAVAIR Systems Command, Washington, DC

AN OPTICAL-FIBRE MULTI-TERMINAL DATA SYSTEM FOR AIRCRAFT 763
J. G. Farrington, Standard Telecommunication Labs Limited, Essex, United Kingdom

THE IMPACT OF FIBER OPTIC MULTIPLEXING ON DISTRIBUTED AVIONICS ARCHITECTURE DESIGN, James M. Bain, AFAL, W-PAFB, OH 783

FIBRE OPTIC REGENERATIVE MULTI-ACCESS NETWORK 793
Robin D. Beasley, Marconi Avionics, United Kingdom

FIBER OPTIC DATA BUS SYSTEM DESIGN CONSIDERATIONS 803
L. Faltzer, S. Cole, and R. Mastinet, ITT

AN AIRCRAFT 1553B COMPATIBLE FIBER OPTIC INTERCONNECT SYSTEM 805
Robert Betts, IBM, Owego, NY

MIL-STD-1553B, 21 September 1978 823

# MULTIPLEX DATA BUS CONFERENCE

10-12 OCTOBER 1978

DAYTON, OHIO

## SESSION I

## CURRENT DATA BUS APPLICATIONS

MODERATOR: John R. Cannon
ASD/XRE
Wright-Patterson AFB. Ohio

# F-16 MULTIPLEX: A SYSTEMS PERSPECTIVE

D. E. Sundstrom, W. B. Anderson, Jr., and S. A. Alford
General Dynamics Fort Worth Division
Fort Worth, Texas 76101

## ABSTRACT

The F-16 avionic system is integrated via a MIL-STD-1553 serial digital multiplex data bus. The architecture provides a primary and backup controller within a complement of nine subsystems. This paper addresses the control concept and implementation of the data bus in the F-16 and provides an assessment of performance. The control concept centers on the criteria to be used for access to the bus, error handling procedures, and the use of channel redundancy. The software-implemented controller has proven efficient in design (950 16-bit words, under 2% duty cycle) and effective in error control (data loss rate of zero over a 4-hour period). Analysis of flight test data reveal that systematic sources amount for virtually all data bus errors found during flight testing.

## I. INTRODUCTION

The F-16 initial avionic architecture was evolved in the 1973-1975 time frame; in this period a commitment was made to implement interface data transfers by means of a dual-redundant, serial digital, multiplex data bus. Choice of the time-division multiplex bus and conformance to the then evolving military standard[1] offered the best potential for lightweight, reliable communications and long-term equipment growth compatibility.

In the definition phase, critical features of the avionic architecture were settled: each sensor would contain its own specialized processing, backup modes of operation were defined, and a separate fire control computer would be provided to integrate sensor data and pilot commands into a coordinated fire control solution [2,3].



PRECEDING PAGE BLANK

The multiplex controller philosophy was settled by opting for a simple and unambiguous structure: the fire control computer, when operating, would be the bus controller; if the fire control computer was not operating, the inertial navigation unit would assume bus control. In addition, the fire control computer would never function as a terminal. These decisions enabled interfaces to be specified and contracts to be well-defined. With this architecture and system philosophy, the full scale development was initiated in early 1975.

The detailed design of the bus control algorithm in the primary bus controller, the fire control computer, proceeded from the premise that "ability to communicate" was the criterion to be used in redundancy management, i.e., selection of bus channel and in sensing the on/off/failed status of equipment on the bus. The backup control algorithm, as implemented in the inertial navigation unit, was considerably simpler, for two reasons. First, the number of data blocks to be managed was comparatively small, and second, the fault-reporting and error-recovery procedure requirements were considerably reduced. In this paper, we address the primary bus control algorithm implemented in the fire control computer, with mention made of the backup control algorithm in appropriate areas.

Although the F-16 is the first aircraft to employ a MIL-STD-1553 compatible multiplex system, it is not the first aircraft to use a data bus. The F-15[4], the space shuttle[5], and the B-1[6] all utilize 1 MHz, serial digital, time division multiplex buses. All of these vehicles preceeded the publication of the initial version of the military standard for such buses. More recently, the Air Force's DAIS project has utilized the successor standard, MIL-STD-1553A, to integrate communications among its elements.[7,8]

The military standards for multiplex systems, while addressing protocols and message formats, do not address the mechanization of the control algorithm. Hence, there is very properly room for considerable variation in implementations. The F-16 implementation, described herein, is

characterized by low processor overhead (CPU duty cycle and core utilization), adaptive configuring of data transfers to reflect the communicative status of each subsystem, and efficient and effective error handling.

The paper addresses this implementation by summarizing relevant features of the F-16 avionic architecture in Section II, the bus control concept and interface strategy in Section III, and the primary controller implementation in Section IV. In Section V an analysis of performance based on flight test data is provided. Section VI contains a perspective view of the controller implementation and suggests testing approaches to identify and reduce systematic error sources.

## II. F-16 AVIONIC ARCHITECTURE

The organization of data flows within the F-16 architecture may be viewed in several ways. The physical interface (Fig. 1) illustrates the electrical interconnections. The single discrete line shown provides the bus control hand-off cue between the primary controller (FCC) and the backup controller (INU). The functional interface (Fig. 2) illustrates the effective data flows to the various equipments. The labels identify the data blocks which are transferred between elements; each data block and its content are specified in a formal interface control document. A summary of the characteristics of this multiplex traffic is provided in Table 1.

The architecture of the F-16 avionic system reflects basic design and procurement strategies, both of which influence the design and operation of the data intercommunication system. An underlying architectural concept was that each sensor should do its own local processing; this results in low data rates on the bus as only processed data is output by a sensor. This same concept results in minimal data latency as transport lags caused by intermediate processing are eliminated. A further result, useful in procurement and development, is that each subcontractor has a well-bounded task so that demonstration

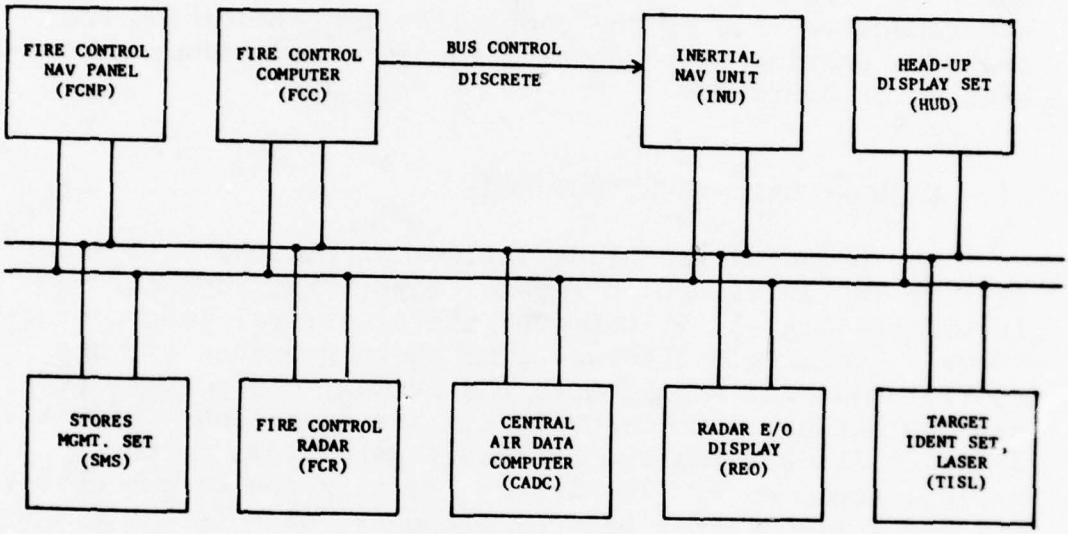



Figure 1 F-16 MULTIPLEX SYSTEM ARCHITECTURE

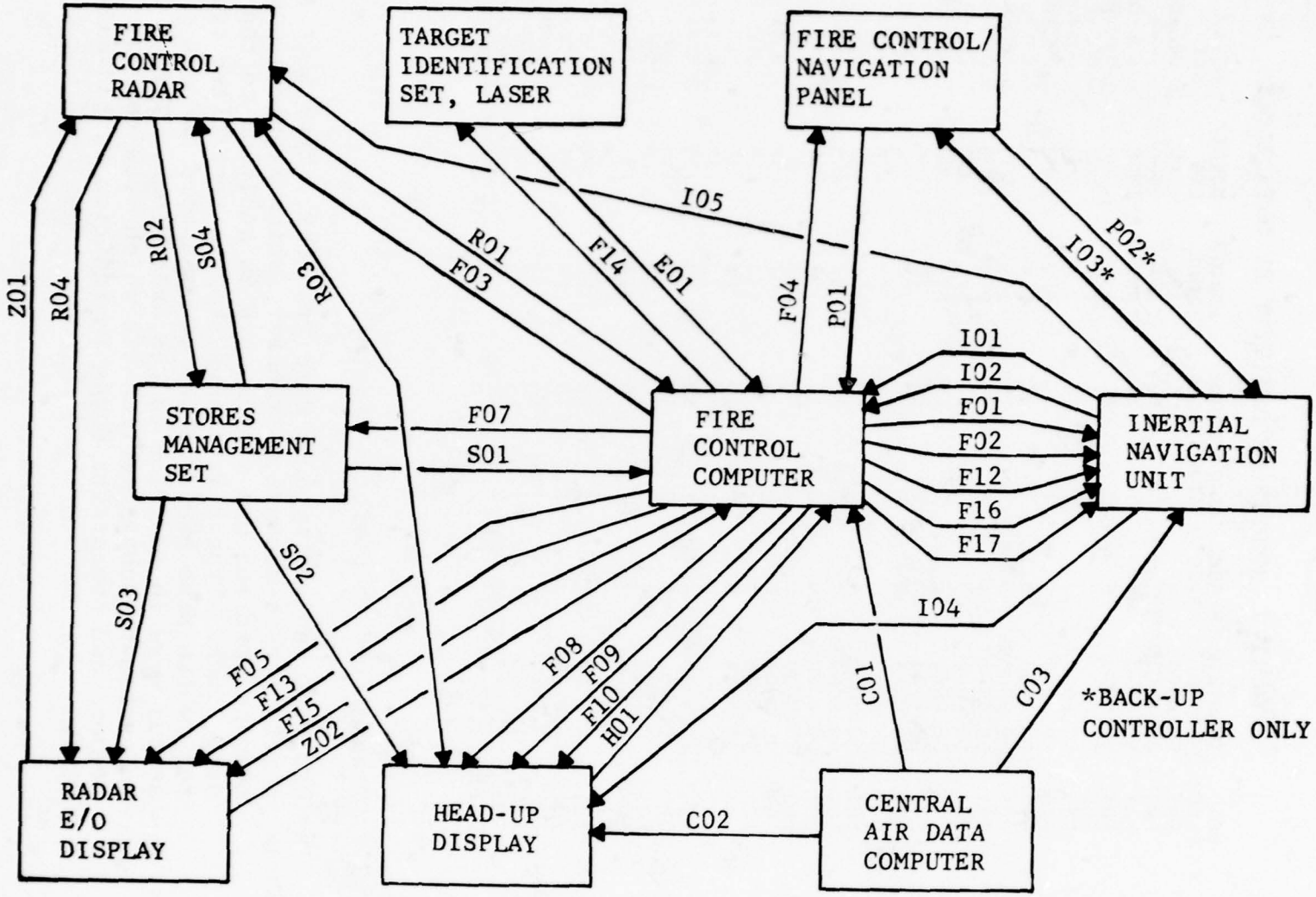



Figure 2 F-16 FUNCTIONAL INTERFACE DATA ROUTINE

## TABLE I MULTIPLEX ACTIVITY BY DATA BLOCK

| DATA BLOCK | TRANSFER RATE | BLOCK SIZE | CMD TYPE | WORDS/ BLOCK | WORDS/ FRAME | WORDS/ SEC | BITS/ SEC |
|---|---|---|---|---|---|---|---|
| E01 | 50.0000 | 4 | 2 | 6 | 192 | 300 | 6.00E+03 |
| F02 | AS REQ'D(*) | 16 | 1 | 18 | --- | --- | --- |
| F03 | 50.0000 | 24 | 1 | 26 | 832 | 1300 | 2.60E+04 |
| F05 | 50.0000 | 4 | 1 | 6 | 192 | 300 | 6.00E+03 |
| F07 | 50.0000 | 2 | 1 | 4 | 128 | 200 | 4.00E+03 |
| F09 | 50.0000 | 18 | 1 | 20 | 640 | 1000 | 2.00E+04 |
| F10 | 50.0000 | 24 | 1 | 26 | 832 | 1300 | 2.60E+04 |
| F14 | 50.0000 | 5 | 1 | 7 | 224 | 350 | 7.00E+03 |
| H01 | 50.0000 | 9 | 2 | 11 | 352 | 550 | 1.10E+04 |
| I01 | 50.0000 | 28 | 2 | 30 | 960 | 1500 | 3.00E+04 |
| I04 | 50.0000 | 13 | 3 | 17 | 544 | 850 | 1.70E+04 |
| I05 | 50.0000 | 13 | 3 | 17 | 544 | 850 | 1.70E+04 |
| R01 | 50.0000 | 21 | 2 | 23 | 736 | 1150 | 2.30E+04 |
| R02 | 50.0000 | 3 | 3 | 7 | 224 | 350 | 7.00E+03 |
| R03 | 50.0000 | 5 | 3 | 9 | 288 | 450 | 9.00E+03 |
| S02 | 50.0000 | 4 | 3 | 8 | 256 | 400 | 8.00E+03 |
| S04 | 50.0000 | 4 | 3 | 8 | 256 | 400 | 8.00E+03 |
| C01 | 25.0000 | 10 | 2 | 12 | 192 | 300 | 6.00E+03 |
| C02 | 25.0000 | 9 | 3 | 13 | 208 | 325 | 6.50E+03 |
| C03 | 25.0000 | 2 | 3 | 6 | 96 | 150 | 3.00E+03 |
| F13 | 25.0000 | 7 | 1 | 9 | 144 | 225 | 4.50E+03 |
| R04 | 25.0000 | 9 | 3 | 13 | 208 | 325 | 6.50E+03 |
| F01 | 12.5000 | 7 | 1 | 9 | 72 | 113 | 2.25E+03 |
| F04 | 12.5000 | 8 | 1 | 10 | 80 | 125 | 2.50E+03 |
| I02 | 12.5000 | 10 | 2 | 12 | 96 | 150 | 3.00E+03 |
| P01 | 12.5000 | 4 | 2 | 6 | 48 | 75 | 1.50E+03 |
| F08 | 6.2500 | 28 | 1 | 30 | 120 | 188 | 3.75E+03 |
| F15 | 6.2500 | 29 | 1 | 31 | 124 | 194 | 3.88E+03 |
| S01 | 6.2500 | 31 | 2 | 33 | 132 | 206 | 4.13E+03 |
| Z01 | 6.2500 | 1 | 3 | 5 | 20 | 31 | 6.25E+02 |
| F12 | 1.5625 | 25 | 1 | 27 | 27 | 42 | 8.44E+02 |
| F16 | 1.5625 | 29 | 1 | 31 | 31 | 48 | 9.69E+02 |
| S03 | 1.5625 | 2 | 3 | 6 | 6 | 9 | 1.88E+02 |
| Z02 | 1.5625 | 1 | 2 | 3 | 3 | 5 | 9.38E+01 |

* Transmitted on position update

** CMD TYPE denotes type of data transfer. Type 1 is a controller-to-subsystem transfer, which requires one command word and one status word in addition to the data block. Type 2 is a subsystem-to-controller transfer, which also requires one command word and one status word in addition to the data block. Type 3 is a subsystem-to-subsystem transfer which requires two command and two status words in addition to the data block.

of performance and problem allocation in development are clear to all parties.

All equipments at present operate to asynchronous clocks, although provisions were made to directly synchronize the FCC and HUD processors. Since it was thought in initial design that synchronization (or at least periodic time alignment) might be needed, the controller was provided with a capability to command all terminal clocks to be reset simultaneously. This capability is not presently used. In the present implementation, the truly time-critical data comprises inertial platform measurements. These data are transmitted with a time-tag so that users may establish the latency of this data.

Overall, the benefits of asynchronous processing are primarily simplicity in overall control strategy and interface, whereas the disadvantage is the introduction of variable (random) data latency, depending upon relative phasing of equipment clocks. In applications to data this variable latency has not proved to be a problem. In the few areas where latency effects can be noted (primarily in dynamic displayed data) the latency can be removed by appropriate use of time-tag or by algorithmic compensation.

## III. BUS CONTROL CONCEPT AND INTERFACE STRATEGY

There are two keys to any bus control algorithm design: (1) the predominant transmission mode, whether periodic or on demand, and (2) the error-handling procedures. In the F-16, a predominantly periodic interface definition was pursued. This greatly facilitated the partitioning of functional requirements among contractors and minimal contractor-to-contractor interface sensitivities. It also enabled a computer program structure which takes advantage of scheduled inputs and outputs to minimize data latency. The efficient design of error-handling procedures is critical to the overall system operation. Error-handling in the primary controller was implemented in the flight program software. Here again, simplicity provided to be

the key that assured low data losses without the considerable overhead that could be consumed by elaborate recovery procedures.

The bus control and interface design principles are both few and simple; the understanding of these principles leads directly to an effective implementation.

Ability to communicate is criterion for access to bus. Early discussions focused on the proper criteria for access to the bus, that is, when should the controller initiate communications with a terminal? Some proposed solutions relied on status discretes from each subsystem; receipt of such a discrete would indicate readiness to communicate. Ultimately, the discretes were included in the physical wiring, but are used only for fault reporting, and not in any way utilized in the bus control algorithm. The algorithm determines communication status assessment through periodic polling of each terminal. Based on the results of polling, the data communications to a subsystem are either established or deleted. If a subsystem responses to a poll, full data communications are immediately established in the controller's current command table. On the other hand, if a subsystem fails to respond to a poll for two consecutive polling periods, the appropriate data transfer commands are deleted from the controller's current command table. This periodic polling continues so that subsystem communication status changes may be discerned, without reference to any additional input.

Maintaining communication is criterion for use of the redundant channel. Each subsystem is connected to both channels of the dual bus system, Bus A and Bus B; the controller can select either channel. However, both channels may not be in use simultaneously. Obviously, many strategies for channel selection could be implemented. The strategy selected is one of the simplest: The controller uses the channel that worked last time. For example, if a transfer was attempted on Bus A and was successful, the next transfer of the same block would also be on Bus A. Alternatively, if an attempt on Bus A failed (error occurred) then a retry of that transfer would be attempted on Bus B. Consequently, communications with a subsystem will migrate to the channel that is functioning.

Traffic flows are constant rate. The interface implemented on the F-16 calls for fixed size data blocks transferred at predetermined rates. Once communication is established, the appropriate data blocks are transmitted. A few data blocks are specified to be transmitted on an 'as required' basis; of these all but one have been placed on a fixed, slow-rate basis. The remaining 'as required' block contains inertial system position update data and is only transmitted when a position update is accomplished. Changes of mode or event occurrence do not affect traffic rates, but only the data content of the affected blocks.

Error handling must be efficient. In any controller design the method of error handling is a primary concern. The objective of error handling is to provide the communication robustness needed in real-world environments. Errors will occur; the error handling scheme must be an integral part of the bus management design. Here again, simplicity prevails. The method of error handling implemented on the F-16 follows this procedure: (1) detected error causes software interrupt, (2) software marks a log of errors and either retries the command on the alternate bus or skips the command. The command is skipped only when the retry on the alternate bus failed. This method may be summarized as 'fail once, retry on alternate, fail twice, go to next command.'

Since error handling is so important, it is worth noting that several strategies were examined. In particular, a strategy which called for examining, in real-time, the cause of the error and then deciding action was considered and discarded. The reason is that such a strategy requires excessive computing duty cycle to implement and yields very little in improving communications reliability.

In our implementation the retry is limited to once per command (per scan of the command table) and is initiated on the alternate bus. The retry was selected to be on the alternate bus in favor of the initial bus on the basis of likely error source. It seemed reasonable (and has been justified by experience as shown in Section V) that most errors would not arise from random noise corruption of a

data block, but rather from systematic causes. Candidate causes would include timing lock-outs in the terminal hardware, broken or intermittent wiring, faulty coupling matrices, or EMI. Such causes would be likely to affect one bus more than the other - hence the alternate bus retry strategy and the consequent migration of traffic to the working channel.

## IV. CONTROLLER IMPLEMENTATION

The primary bus controller is implemented in the software of the fire control computer. This computer is a Delco M362F machine procured for the F-16 and programmed by General Dynamics. The Delco M362F computer includes a bus control communication processor which operates under software control. This processor is known as the Serial Data Interface (SDI). The SDI, once given a start command, will read and execute complete bus transfer sequences as defined in the command table. Error detection is also provided by the SDI. The software involvement comprises (1) initiating SDI start commands, (2) providing error interrupt handling, and (3) analyzing poll responses.

Each of the commands (stored in main computer memory) cause the SDI processor to either initiate data transmissions or to perform an internal function such as link (command sequence branch), internal self-test, or stop. Software intervention is required only to start the SDI processor and to respond to SDI initiated interrupts. Any bus transmission which fails to complete successfully causes a "command failure" interrupt to be generated to the CPU. In addition to this, any SDI command may be designated so that a "command task complete" interrupt can be generated when this command is executed by the SDI processor. This forced interrupt is used twice per FCC minor frame (1) to signal the FCC software when key input data have been received and (2) when all scheduled transfers for the current time frame have been completed. The SDI processor is commanded to initiate a command sequence once per 20 millisecond time frame (corresponds to 50 Hz rate).

The avionic interface is structured into data blocks whose transfer rate is 50 Hz or lower, where lower rates are powers of 2 submultiples. The lowest data rate is 1.5625 Hz. At present, no data is transferred at the 3.125 Hz rate. Within this structure, the command sequence in each 20 millisecond minor frame interval consists of data blocks from the 50 Hz group plus those blocks from one of the submultiple groups. Since the lowest rate is 1.5625 Hz, it requires 640 milliseconds (defined to be a major frame) to complete a full data transfer sequence.

To implement this periodic transfer structure, the SDI command table (Figure 3) is linked prior to the start of each minor frame. This is accomplished by setting two link commands to provide the desired path through the command table. The command sequence has been structured to minimize data latency relative to the processing which is being executed concurrently by the CPU. The content of the command table is modified by the results of poll processing to contain transfer-commands for only those subsystems which are actively communicating.

Polling of the subsystems by the controller is used to selectively configure the command table, deleting transfers when the subsystem is not communicating and restoring transfers when polling response indicates an active subsystem. Each subsystem is polled on each channel with a dedicated function command. This command requests the addressed terminal to respond with its current status word. Any poll command which fails (either due to a transmission failure or a fault status reported by the addressed system) results in a CPU interrupt, and software is provided to record the poll command failure. The results of the subsystem poll are filtered with the results of the previous poll and then used to configure the transmission commands in the command table, according to the criteria given in Table II.

When a poll response is obtained from a subsystem which had previously not responded (perhaps powered down), transfers with that subsystem are immediately restored. Hence, on failure or power down of a subsystem, transfers are restored immediately. Polling is accomplished at a

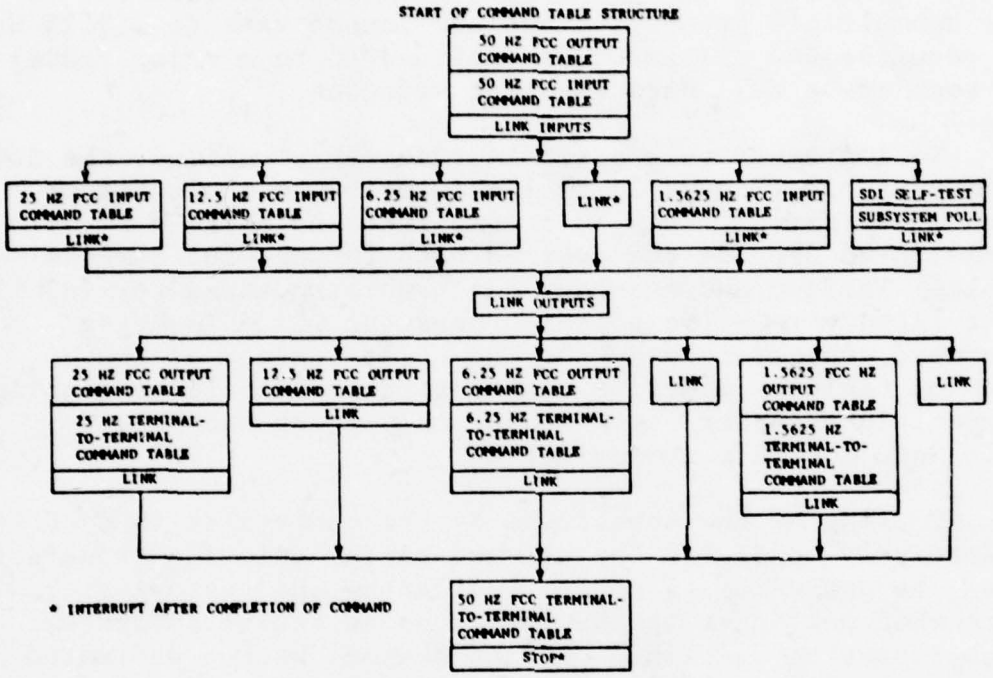



Figure 3 BUS CONTROL COMMAND TABLE STRUCTURE

Table II BUS SELECTION RELATED TO POLL PROCESSING RESULTS

a. Controller-to-Subsystem and Subsystem-to-Controller Transfers

| SUBSYSTEM POLL RESULTS | SELECTED BUS |
|---|---|
| A & B FUNCTIONAL | A |
| A FAIL | B |
| B FAIL | A |
| A & B FAIL | NOTE 1 |

b. Subsystem-to-Subsystem Transfers

| FIRST SUBSYSTEM POLL RESULTS / SECOND SUBSYSTEM POLL RESULTS | A & B FUNCTIONAL | A FAIL | B FAIL | A & B FAIL |
|---|---|---|---|---|
| A & B FUNCTIONAL | A | B | A | NOTE 1 |
| A FAIL | B | B | NOTE 1 | NOTE 1 |
| B FAIL | A | NOTE 1 | A | NOTE 1 |
| A & B FAIL | NOTE 1 | NOTE 1 | NOTE 1 | NOTE 1 |

NOTE 1. Bus A selected if first occurrence of double fault or crossed-bus fault; on second occurrence, the affected data blocks are deleted from the command sequence.

1.5625 Hz rate and the results are processed immediately following poll completion.

We draw a distinction between bus management and fault reporting for avionic maintenance. As can be seen, the ability to communicate criterion is sufficient for bus management. The polling results, of course, can also be used in assessing the need for, or target of, maintenance actions. We accomplish this by accumulating the poll results in a fault-reporting software routine. This routine combines the poll failures with power on and installation status to distinguish between true failures and power-off-or not-installed failures. Those results representing true failures are then made available to the pilot and maintenance personnel.

In addition to poll response errors, interrupt software is provided to respond to data transmission errors. The error handling software indexes an error response table, which contains an entry dictating the appropriate error response for each command. For data transmission failures, the error response table indicates (1) whether the command is to be retried, (2) the bus to be used for the entry and, (3) if a data storage buffer in FCC memory should be marked as invalid. In current implementation, all data transmissions are selected for immediate retry on the alternate bus. However, in future use it may be desirable to selectively retry commands or to retry on the same bus for subsystems having only one channel. The table driven nature of the interrupt handler has provided the desired flexibility in error response without an unnecessary burden on the CPU.

The resulting implementation utilizes minimal memory and duty cycle. The control algorithm consists of (1) normal commands to start the bus, (2) an interrupt handler to cue the selection of command links, (3) an error interrupt handler, (4) a module to analyze poll results, and (5) the bus command tables. The total memory utilization is 950 16-bit words, while the CPU processing utilizes less than 2% of the machine duty cycle. Error rates are sufficiently low (see Section V) that error handling adds

PRECEDING PAGE BLANK

negligible processing overhead. Typical error response times are 118 microseconds for first failure of command, 122 microseconds for second failures, and 93 microseconds for a poll failure.

## V. PERFORMANCE

The first flight of an F-16 with an operating multiplex bus aboard was on 12 August 1976. Since that date, over 500 avionic flights have been made. The accumulated experiences of the flight test environment demonstrate clearly the robustness of the controller design in response to extreme ranges of error and fault conditions. In this section, we describe qualitatively some of the anomalous conditions encountered and also provide a quantitative analysis of in-flight error rates.

The analysis of in-flight events is made possible through a flight test instrumentation system which can eavesdrop on both buses. The instrumentation system can record up to 90 minutes of multiplex data on multi-channel magnetic tape. A ground based computer is used for post-flight processing of data recorded by the instrumentation system. The data processing system can provide either raw data or can selectively extract information from the total data set. For example, a computer report of only those messages to and from a particular system may be prepared or a report of only incomplete or erroneous messages may be made.

To further provide performance data, two error counters were implemented as an integral part of the controller software. Each time a particular message must be retried, due to any type of error, the single-error counter is incremented. Each time a particular message fails to be transacted properly on the retry, the double-error counter is incremented. These two counters provide an accurate measure of data block error rates. Subsequent analysis of raw data at the time one or both of the counters is incremented can indicate the specific nature of a failure, e.g., not enough words transmitted, no status response, status reponse with error bit set, etc.

In terms of qualitative performance, the flight test environment has provided a considerable variety of instances in which controller response is of interest. In each of the situations described below the controller operated as designed, providing a maximum degree of communications integrity.

- o Failed terminal on the primary bus. Fault detected by controller, appropriate subsystem data switched to alternate bus. No loss of communications. Fault reported to maintenance crew through self-test software.

- o Failed terminal on alternate bus. Fault detected by analysis of poll responses. Data remains on primary bus. No loss of communication. Fault reported to maintenance crew through self-test software.

- o Double bus failure to one subsystem. Loss of subsystem communication; retries unsuccessful. Poll responses analyzed and faulted subsystem isolated from data stream. Period i.e., polling maintained to assess subsystem status. Self-test software verifies that subsystem power is on prior to reporting the fault to the pilot.

- o Intermittent double bus failure to one subsystem. Retries generally unsuccessful. Loss of communication if failure persists through two polling periods, resulting in isolating subsystem from data stream. Communications re-established as soon as response to poll is obtained. If subsystem power is on, self-test software reports fault to pilot, identifies as intermittent.

- o Mismatched switching matrix (prototype). Resulting high noise and degraded signal

waveform created high error rate situation. Retries successful to the point that no external performance degradation was discernable. False reports of subsystem bus failures together with garbled instrumentation data led to the identification of the problem and its source.

- o Concurrent bus traffic on a single bus. Arose when terminal wrongly interpreted a specific data word to another subsystem as its terminal address and initiated a data output. Controller detected event as an incorrect word count error, shifted erred block to alternate bus and retried. Retry successful because timing lockout in faulty terminal inhibited response. Next regular transmission of this block resulted in same error, overcome by same process. *No loss of communication.* Problem discovered by post-flight instrumentation data analysis.

- o Two terminals respond to the same address. Arose when a broken wire in one terminal resulted in an effective change of terminal address. Unfortunately, another terminal also owned that address. Faulty terminal then competed with rightful terminal in acknowledging commands and transmitting data. Poll analysis identified double bus failure to faulty terminal. Strange in-cockpit behavior of faulted subsystem made problem evident to ground crew.

- o One terminal could not accomplish two back-to-back transmissions on the same bus. A design flaw in early hardware associated with this subsystem resulted in locking out transmission of a data block which immediately followed transmission of a somewhat longer data block. Retry of the second

block was successful. The next sequential transmission was initiated (as normal) on the A bus and was successful. The result was that this subsystem's outputs were channeled in an A, B, A sequence. No communication loss, no fault reported. The problem was discerned and identified through post-flight error analysis.

- o Inconsistent status word content. An early design flaw in one subsystem lead to a situation in which the status word produced by this subsystem had content which varied depending on the type of command received. Specifically, if a dedicated function command was received, the terminal hardware immediately responded with a canned, valid status word. If the command involved data, however, the content of this status word was read from the CPU. The trouble arose when the subsystem faulted and set the terminal status bit bad. This resulted in double errors on all retries, and hence, total loss of data communication with this subsystem. At poll time, however, the terminal would respond as healthy. It was clear from in-cockpit behavior that the subsystem was effectively dead, yet the avionic self-test fault reporting did not indicate the suspected double bus failure. On-line analysis with an oscilloscope solved the mystery.

To provide a quantitative assessment of performance we have evaluated error rates on four recent F-16 flights. The airplanes were F-16A No. 3 and F-16B No. 1; both were built and flown as part of the F-16 full-scale development program. The instrumented airplanes had the same complement of avionics equipment, consisting of full-scale development and pre-production hardware. Of the nine

possible subsystems, only the Target Identification Set was not installed. The data rates expressed in Table III represent the multiplex traffic generated by this avionic configuration.

The error data summarized from the analysis is provided in Table IV. The conditions we required in selecting analysis intervals were (a) all installed subsystems operational (b) the airplane be airborne. Within these constraints we extracted the intervals displayed in the Table. Since a typical flight is of one hour airborne duration, these intervals represent the steady-state multiplex performance. In addition, no intervals were selectively discarded to eliminate errors. The above two criteria were the only selection criteria.

Two features immediately stand out from inspection of the Table. First, the obvious difference in single-error counts between the two airplanes. Second, the low number of double-error counts experienced on all flights. Taken together, these features indicate the strong reliability of the avionic multiplex system. Summing the count of double errors together with the flight durations on all four flights, we obtain the result that the data loss rate was zero over a period of 273 minutes.

Returning to the differences between the single-error rates on the two aircraft, a detailed analysis of error distribution in time as well as inspection of raw transaction data provided insight into the error source.

A once-per-second report of the bus controller error counters during flight 241 of F-16A No. 3 revealed that the retry counter was incremented nearly once each printout, i.e., the errors appeared "uniformly" distributed across the entire sample period. (The computed average error rate was actually 0.79 errors per second.)

Analysis of raw data from selected time intervals disclosed the presence of a highly systematic error condition. Ten message anomalies which caused the error counter to be incremented were isolated. In nine of the ten cases

Table III FLIGHT TEST DATA RATES

| | | |
|---|---|---|
| Date Block Rate: | 956.25 | blocks per second |
| Safe Block Rate: | 382.8125 | blocks per second (Note 1) |
| Total Block Rate: | 1339.0625 | blocks per second (Note 2) |
| Total Word Rate: | 14,405 | words per second (Note 3) |

NOTES

1. Safe blocks consist of a dedicated function command and the associated status word response. These blocks are inserted into the command streat following every subsystem-to-subsystem transfer. They assure that the receive in such a transfer will be reset in the event that the transmitter did not send.

2. Represents the total data stream on which errors are counted; polling blocks are excluded from the error count.

3. Includes command, status, and data words associated with data and safe blocks; polling blocks excluded.

examined, one particular subsystem failed to respond with a status word after being commanded to receive a seven word data block by the controller. The retry on the alternate bus was successful in every case. Further investigation disclosed that the status response error occurred only when the received command, which is executed once each 80 milliseconds, followed a controller polling sequence, which occurs only once each 640 milliseconds. The interaction of the polling sequence with the receive command clearly "induced" the predominate error condition observed on F-16A No. 3. (The other anomaly discovered in the raw data was a failure of another subsystem to respond to a command to receive two data words from the controller. Again, the retry on the alternate bus was successful.)

These data provide insight into the typical types of errors which have been experienced on the F-16 multiplex system. The overwhelming majority of errors have been systematic, attributable primarily to terminal design flaws. Within this category, the most common error has been failure of a receiving terminal to acknowledge receipt of data, i.e., no receiver status word was transmitted. On the other hand, we have yet to find a single error which clearly represented a signal corruption by background noise. These factors lead to thoughts about multiplex improvements, discussed in the following section.

Returning to the data of Table IV, we have calculated two error rates associated with each flight. Both the word and bit error rates must be interpreted as equivalent error rates, since the errors actually counted are on a block basis. That is, we assumed that a block error is equivalent to a single word or single bit error, which is not necessarily true. In fact, the true bit error rate is considerably better than the data would indicate, simply because we know that most of the errors are associated with protocol defaults (missing status words).

## VI. SYSTEM DESIGN PERSPECTIVES

The distillation of the F-16 multiplex system design and evaluation experiences leads us to two perspectives.

Table IV FLIGHT TEST RESULTS SHOW LOW ERROR RATES

| F-16 | FLIGHT NUMBER | TYPE OF FLIGHT | ANALYSIS DURATION (MIN:SEC) | ERRORS | | BUS TRAFFIC | | ERROR RATES | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | SINGLE | DOUBLE | NO. WORDS PER SEC | NO. BITS PER SEC | PER WORD | PER BIT |
| B1 | 185 | Air-Ground Evaluation | 80:54 | 2 | 0 | $6.7X10^{7}$ | $1.3X10^{9}$ | $3.0X10^{-8}$ | $1.5X10^{-9}$ |
| B1 | 186 | Air-Air Evaluation | 67:36 | 1 | 0 | $5.6X10^{7}$ | $1.1X10^{9}$ | $1.8X10^{-8}$ | $8.9X10^{-10}$ |
| A3 | 241 | Air-Air Evaluation | 69:46 | 3246 | 0 | $5.8X10^{7}$ | $1.2X10^{9}$ | $5.6X10^{-5}$ | $2.8X10^{-6}$ |
| A3 | 247 | Air-Air Evaluation | 54:50 | 2567 | 0 | $4.6X10^{7}$ | $9.1X10^{8}$ | $5.6X10^{-5}$ | $2.8X10^{-6}$ |
| AGGREGATED RESULTS | | | 273:06 | 5813 | 0 | $2.3X10^{8}$ | $4.5X10^{9}$ | $2.6X10^{-5}$ | $1.3X10^{-6}$ |

First, the design principles as described in Section II provide a sound basis for bus controller implementation. Second, the systematic error modes that dominate noise errors suggests that specific testing beyond that required by the multiplex standard should be explored. A brief expansion of these perspectives is provided in this section.

Our bus control concept focused on maintaining communication among the subsystems. The implementation of this concept required but minimal core-the 950 words include the command tables - and very low execution time. Of equal or greater importance is the efficacy of the error handling implementation. It is not only extremely effective in terms of data block transfer but also accomplished the error handling with negligible processing time.

The transition logic which causes bus controller responsibilities to be transferred between the primary and backup control is simple and effective. It has been exercised extensively in both directions. The avionic system gracefully transitions to and from its backup mode of operation when the bus control logic in the inertial navigation unit is selected and deselected. In the lab, failure of the bus control discrete has been simulated. In this mode, principally because of low duty bus cycle, the system displays can be observed to rapidly alternate between primary and backup display modes. (This condition may be overridden by powering down one of the two controllers.) No unintentional instances when more (or less) than one controller has been active simultaneously have occurred during two years of the flight testing of the F-16.

Our flight test and laboratory experience showed that systematic error modes predominate in the operation of the integrated system. Two important points regarding systematic errors should be made. First, no degradation of avionic system performances was discernible during the flights when systematic errors occurred frequently. This is a testimony to the robustness and spontaneity of the bus control algorithm in providing high communication integrity in the presence of multiple errors. Second, the subsystems which caused the systematic errors had passed

system operator. Despite systematic error conditions, data loss has proved to be negligible (no data loss is over four hours). This implementation of a MIL-STD-1553 multiplex bus has been shown to provide a very satisfactory avionic system communication network.

## REFERENCES


1. MIL-STD-1553, "Aircraft Internal Time Division Command/ Response Multiplex Data Bus," 30 August 1973.

2. England, G. R., "Shop Test Success is a Function of the Airborne System Design," NAECON Record, 1977.

3. Engelland, J. D., "The F-16 Avionic System - Structured for Affordable Performance," NAECON Record, 1977.

4. Susek, J. F., "The Design and Development of the F-16 Avionics Data Bus," AFSC Multiplex Data Bus Conference, November 1976.

5. Mallary, W. E. and Yeo, J. E., "Space Shuttle Multiplex Data Bus System," AFSC Multiplex Data Bus Conference, November 1976.

6. McLaren, B. D., "B-1 Avionics Data Multiplexing System (AMUX)," AFSC Multiplex Data Bus Conference, November 1976.

7. Pensworth, Capt. R. L., "Digital Avionics Information System (DAIS) Multiplex System," AFSC Multiplex Data Bus Conference, November 1976.

8. Mason, R. C. and Slightam, R. J., "Centralized Control in a TDM Data Bus System," NAECON Record, 1977.

# THE APPLICATION OF MIL-STD-1553A TO THE LAMPS MK-III AIRBORNE SYSTEM

Fred Shocket
and
Ralph E. Hungerford

Naval Air Development Center
Warminster, PA. 18974

## ABSTRACT

This paper presents a brief description of the LAMPS MK-III system and its management, and a more detailed description of the MK-III Airborne System. It summarizes the interpretation of MIL-STD-1553A as applied to LAMPS MK-III and describes the software approach to bus controller IO.

## INTRODUCTION

The most evident feature of the Light Airborne Multipurpose System (LAMPS) is the combination of a helicopter and a destroyer. LAMPS is, in a real sense, a destroyer system which dramatically extends the range of the destroyer's sensors and weapons.

The newest version of LAMPS, the MK-III system, is now in full scale development. Successful completion of technical and operational testing of the full scale development system is necessary before the decision can be made to proceed with production. The decision to produce is scheduled for 1981, and the first operational MK-III LAMPS should appear in the fleet in 1984.

The MK-III LAMPS is a integrated ship-air system in which the shipboard and airborne elements communicate over a two way directional digital data link. The normal method of operating LAMPS will require the close coordination of both shipboard and airborne equipment and operators. However, the airborne system and crew can, if necessary, operate independently.

A new and salient feature of the architecture of the MK-III airborne system is the use of digital, command/response, time division multiplexing to interconnect the avionic subsystems; i.e., the use of a 1553A data bus.

The Navy's manager of this system finds himself in an unusual dual role. He is both an Air Project Manager and a Ship Project Manager, and is designated PMA/PMS-266. The system prime contractor is the Federal Systems Division of the IBM Corporation. The SH-60B helicopter is being provided to the Navy by the Sikorsky Aircraft Division of United



PRECEDING PAGE BLANK

Technology Corporation. PMA/PMS-266 has engaged the resources of three Navy activities to conduct independent testing of system software and, later, to provide software maintenance when the LAMPS MK-III is in the fleet. These Navy activities are: the Naval Air Development Center in Warminster, Pa; the Naval Underwater Systems Center in New London, Conn; and the Fleet Combat Direction Systems Support Activity in Dam Neck, Va.

## DESCRIPTION OF THE LAMPS AIRBORNE SYSTEM

The airborne system consists of the seven subsystems illustrated in figure 1:

1. Anti-Submarine Warfare
2. Ordnance Launch Control
3. Communication
4. Data Handling
5. Anti-Ship Surveillance and Targeting (ASST)
6. Navigation
7. Data Display

Two important sensors do not appear in figure 1 since they are not connected to the data bus. They are the Magnetic Anomaly Detection (MAD) sensor and the radar.

Ten Remote Terminals connect these subsystems to the data bus and, thus, to a central computer/bus controller. The LAMPS data bus uses only two of the three types of data transfer allowed by MIL-STD-1553A. Controller to Remote Terminal (RT) transfers and RT to controller transfers are employed. RT to RT transfers are not used. This reduces the software sophistication necessary to control the bus.

Bus control is provided by one of two AN/AYK-14 Standard Airborne Computers used in LAMPS. This central system computer, Standard Airborne Computer 1, also provides the computation and control required to perform all mission related airborne functions and to integrate the avionic equipment into a cohesive system. All LAMPS signal procesisng is performed at the subsystem level.

Standard Airborne Computer 2, a slightly different configuration of the AN/AYK-14, is used as a remote terminal in the ASST subsystem to process Electronic Warfare sensor signals. If Standard Airborne Computer 1 fails, its computer program would be loaded from the Magnetic Tape Memory into Standard Airborne Computer 2. Electronic Warfare would no longer be processed, and Standard Airborne Computer 2 would serve as the bus controller in a degraded system mode. Acoustic Signal Processing is performed in the Advanced Signal Processor (ASP) within the Analyzer Detecting Set. This processor is yet another standard Navy computer. The ASP, once called "PROTEUS", is not connected to the 1553A bus.

There are two 1553A data bus channels in the MK-III air system: Channel 1 and Channel 0. Channel 1 is used as a program load path and a data extraction path, and connects the two AN/AYK-14 computers to the Magnetic Tape Memory. This channel is implemented with redundant busses: Bus A and B. Selection of the bus to be used by Channel 1 is a bus controller software function. Channel 0, consisting of a single bus, is the communication path between the bus controller and RT's of the avionic subsystems. Cost was the primary factor in the decision not to provide bus redundancy for channel 0. Channel 0 is approximately 80 feet long.

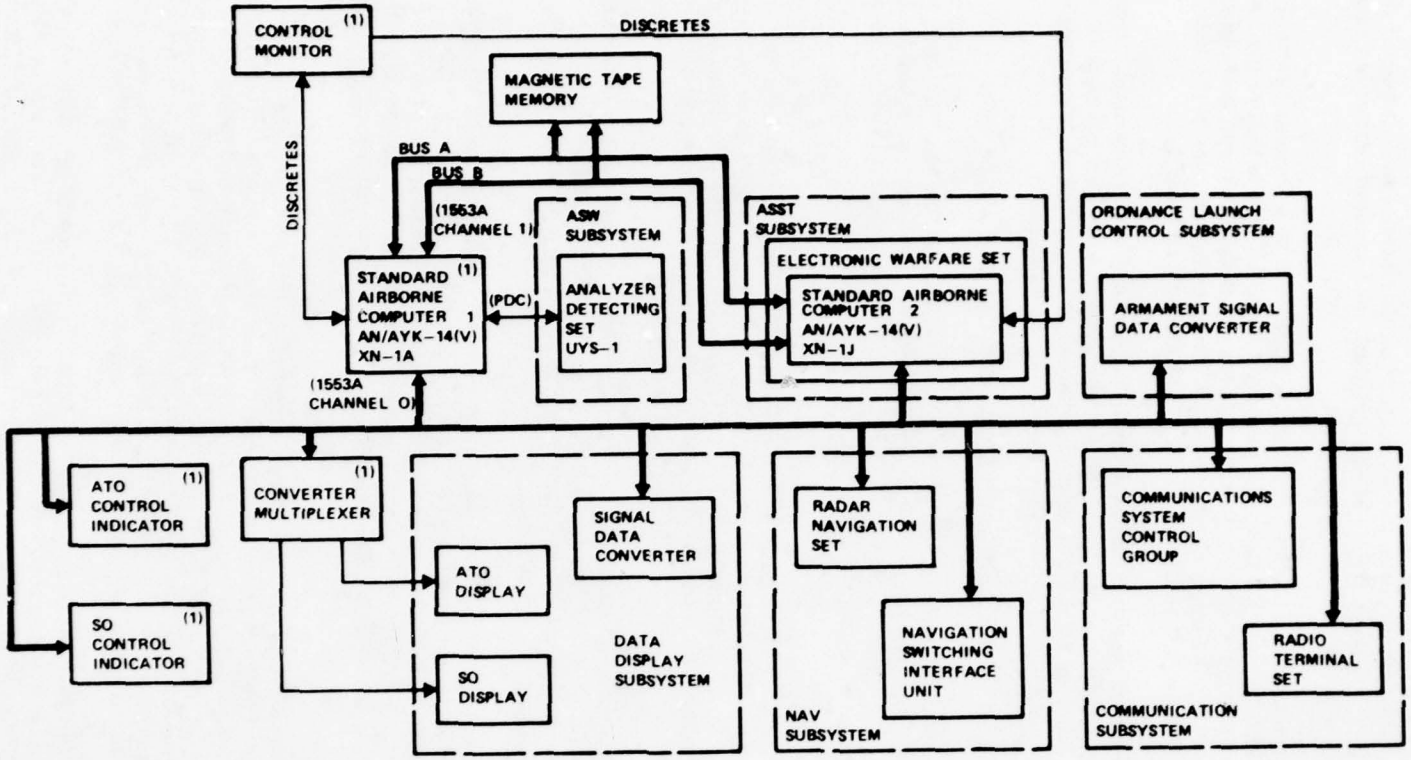



Figure 1 - Computer Interface Block Diagram

The Magnetic Tape Memory was placed on a separate channel so that even if the system bus (channel 0) gets tied up, the computer program for Standard Airborne Computer 1 still can be reloaded and the system reinitialized.

The ASP is connected to Standard Airborne Computer 1 by a Proteus Digital Channel (PDC) rather than by the data bus. The primary reason for this is that at the time the system was designed, the ASP was in the late stages of development, and did not include a 1553A interface. On the other hand, the AN/AYK-14 computer was in the early stages of development, and the incorporation of a PDC interface was feasible.

The remote terminals and Standard Airborne Computer 1 also communicate, in parallel to the 1553A bus, via the discrete channel. This channel provides an independent path by which the computer can issue a master clear signal to any terminal, and provides a method for the remote terminals to report their power on/off status.

## DESCRIPTION OF THE LAMPS APPLICATION OF THE 1553A BUS

IBM has issued a detail specification (IBM 6226114C) which defines the requirements and characteristics of the interface between the LAMPS data bus and an avionic subsystem. It is intended to amplify and interpret MIL-STD-1553A. The standard defines transmission word formats for command, data, and status words.

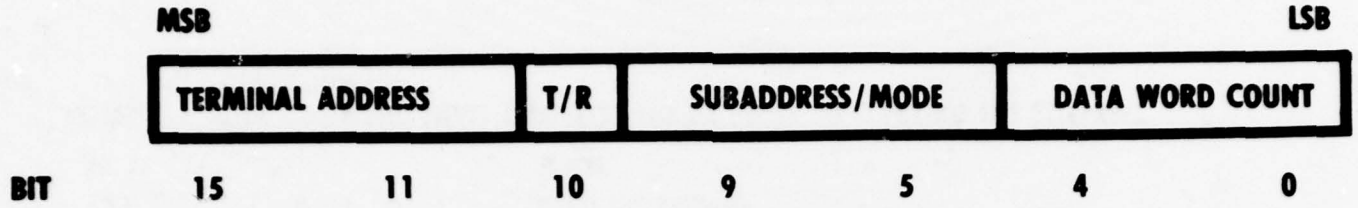
Figure 2 illustrates the command word format. MIL-STD-1553A reserves only one value of the mode control code associated with the Subaddress/Mode (SAM) field (the value 00000). All Subaddress/Mode field values used in the LAMPS system are shown in figure 2. When the Subaddress/Mode contains the value 00000, that implies that the Data Word Count field does not contain a word count. Rather, it contains additional mode control information.

The assignment of the 32 values of this auxiliary mode control field is shown in figure 2. MIL-STD-1553A again reserves the use of the value 00000, in this case to signify dynamic bus allocation. The MK-III system does not employ dynamic bus allocation, but the assignment is included here for completeness.

Where the bus controller has transmitted a command word with the Subaddress/Mode field set to zero and the word count field set to two, signifying a "Transmit Status Word" command, the remote terminal addressed would respond with a Remote Terminal Status Word. Figure 3 shows the LAMPS version of the RT Status Word. It is as specified in MIL-STD-1553A, with the LAMPS assignments of the bits in the Status Code field shown.

If, the bus controller had issued a command word with the Subaddress/Mode field set to two and the word count field set to one (the command to send one "RT Built-in-test Status Word") the RT addressed would respond with an RT Status Word followed immediately by an RT BIT Status Word. The RT BIT Status Word is a uniquely defined Data Word. The structure of an RT BIT Status Word is shown in figure 4. This word provides a method for a Remote Terminal to report specific faults in data bus message traffic, and to provide more detailed information regarding subsystem and subsystem element status.

A Message Error field has been defined, and the significance of bits within this field is shown. When a bit is set in this field, the Remote Terminal will also set the Message Error bit in the RT Status Word. The remaining 16 bits are associated with subsystem faults, and are found in two fields. The information in these fields is uniquely defined for each Remote Terminal. Fault indications in these subsystem fields will cause the Terminal Flag (T/F) bit in the RT Status Word to be set.




**WHERE:**

**T/R FLAG INDICATES THE DIRECTION OF THE DATA FLOW**

**= 0 CONTROLLER TO THE REMOTE TERMINAL**

**= 1 REMOTE TERMINAL TO THE CONTROLLER**

**SUBADDRESS/MODE (SAM) FIELD IS DEFINED AS**

| VALUE | BIT PATTERN (BITS 9-5) | |
|---|---|---|
| 0 | 0 0 0 0 0 — | MODE/DISCRETE DATA |
| 1 | 0 0 0 0 1 — | NORMAL DATA TRANSFER |
| 2 | 0 0 0 1 0 — | TRANSMIT BIT STATUS WORD(S) |
| 3 | 0 0 0 1 1 — | RETRANSMIT LAST MESSAGE |
| 4 | 0 0 1 0 0 — | TRANSMIT LAST COMMAND |
| 5 | 0 0 1 0 1 — | CONTROL COMMAND DATA TRANSFER |
| 6 | 0 0 1 1 0 — | MULTI-MESSAGE TRANSFER |

Figure 2 - Command Word Format

DATA WORD COUNT FIELD IS THE WORD COUNT OF THE NUMBER OF DATA WORDS TO BE TRANSFERRED, EXCEPT FOR USE WHERE THE SUBADDRESS/MODE FIELD IS ZERO. FOR THIS CASE, ONLY THE FOLLOWING DEFINITIONS SHALL BE USED

| VALUE | BIT PATTERN | DEFINITION |
|---|---|---|
| 0 | 0 0 0 0 0 | DYNAMIC BUS ALLOCATION |
| 1 | 0 0 0 0 1 | INITIALIZE TERMINAL |
| 2 | 0 0 0 1 0 | TRANSMIT STATUS WORD |
| 3 | 0 0 0 1 1 | INITIATE SELF-TEST (NOTE 1) |
| 4 | 0 0 1 0 0 | INITIATE PROCESSING |
| 5 | 0 0 1 0 1 | DISABLE BUS A |
| 6 | 0 0 1 1 0 | DISABLE BUS B |
| 7 | 0 0 1 1 1 | NOT USED |
| 8 | 0 1 0 0 0 | NOT USED |
| 9 | 0 1 0 0 1 | NOT USED |
| 10 | 0 1 0 1 0 | NOT USED |
| 11 | 0 1 0 1 1 | NOT USED |
| 12 | 0 1 1 0 0 | INHIBIT STATUS WORD T/F FLAG (BIT 0) |
| 13 | 0 1 1 0 1 | RESET STATUS WORD T/F FLAG (BIT 0) INHIBIT |
| 14 | 0 1 1 1 0 | RESET POWER OFF/ON TRANSIENT INDICATOR |
| 15-19 | | SPARE |
| 20 | 1 0 1 0 0 | INITIALIZE IPL |
| 21 | 1 0 1 0 1 | READ IPL RECORD |
| 22-31 | | SPARE |

NOTES: 1. THIS COMMAND SHALL INITIALIZE THE RT PRIOR TO STARTING THE SELF-TEST OPERATION

Figure 2 - Command Word Format (Contd.)

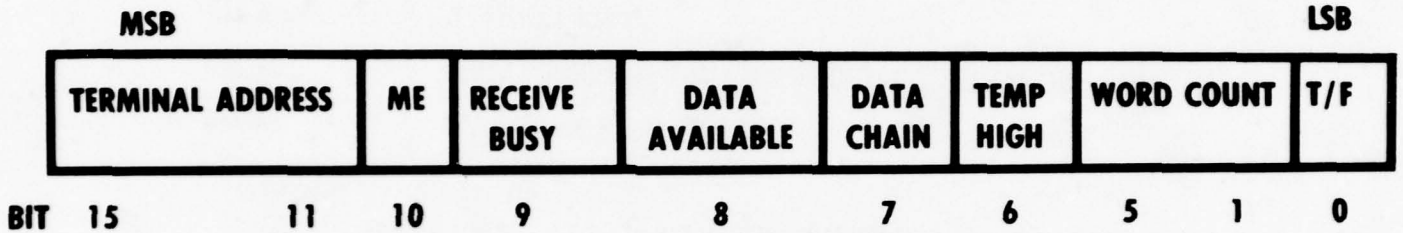
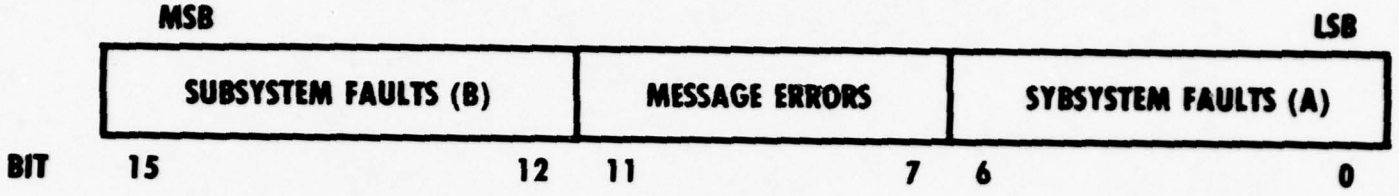



Figure 3 - Remote Terminal Status Word Definition




**WHERE:**

**MESSAGE ERRORS FIELD IS A 5-BIT FIELD CONSISTING OF THE FOLLOWING MESSAGE ERROR FLAGS:**

| | |
|---|---|
| **BIT 11** | **INVALID CONTROL WORD** |
| **BIT 10** | **MULTI-MESSAGE TRANSFER WORD COUNT ERROR** |
| **BIT 9** | **VALIDITY ERROR** |
| **BIT 8** | **COUNT ERROR** |
| **BIT 7** | **INVALID COMMAND** |

**SUBSYSTEM FAULTS (A) AND (B) FIELD SHALL BE DEFINED BY THE INDIVIDUAL EQUIPMENT SPECIFICATION**

Figure 4 - Remote Terminal Bit Status Word

Since the RT BIT Status Word is a Data Word, the 1553A scheme permits as many of these words as is required. The number of BIT Status Words varies from RT to RT, and depends on the amount of fault information associated with the equipment connected to the RT. As currently designed, the greatest number of BIT Status Words that the bus controller sees in response to a BIT Status command is four.

Figures 5, 6, and 7 show representative sequences of commands used in LAMPS. Initialization, Data Transfer and Self Test sequences for the airborne data link terminal (Radio Terminal Set) are illustrated. In the Initialization and Data Transfer sequences the Control Command Data Transfer Command Word is employed to transfer a Data Word containing control information for the data link. This information includes the selection of the forward or aft data link antenna, and further, tells that antenna the relative bearing at which to point in order to aim at the destroyer.

The computer program used in Standard Airborne Computer 1 during a LAMPS mission is called the Avionics Operational Program (AOP). The AOP includes an algorithm that controls the periodic communication between the computer and the remote terminals. Figure 8 shows the polling frequency and I/O rates for each of the RT's. The I/O protocol between Standard Airborne Computer 1 and any RT is based on a 40 millisecond cycle. It can be seen from figure 8 that each device on channel 0 has a polling rate which is a multiple of forty milliseconds. Each 40 millisecond cycle is subdivided into a polling period and a data transfer period. The polling period occurs at the start of the cycle and involves transmission of a transmit status word command to each RT scheduled to be accessed during this particular 40 millisecond cycle. During the polling sequence, the RT Status Word will be examined for the state of bits controlling subsequent data transfers to and/or from each RT. After completion of polling operations, data transfer will commence. During the data transfer period, each RT scheduled for data transfer activity will be commanded to send and/or receive data from Standard Airborne Computer 1. After completion of all data transfer activity, the data bus will remain in a quiet state until the start of the next 40 millisecond cycle.

The number of data transmit commands is thereby reduced to the number of remote terminals that actually have data available for transmission. Simulation studies performed by the prime contractor indicate that the average data bus utilization is 17 percent of capacity, and that the peak load is 29 percent of capacity.

All LAMPS avionic subsystems that connect to the 1553A bus contain the remote terminal within the subsystem. This approach simplifies the management of interfaces, and permits clear definition of responsibility for subsystem performance. The supplier of a subsystem is responsible for the proper operation on the data bus of that subsystem.

In the MK-III system, stubs are connected to the bus as shown in figure 9. This is in lieu of a coupler box. The stub splices are protected by the backshell of the connector. Coupling boxes were avoided in order to reduce weight and to reduce the reliability risk by reducing the number of connectors.

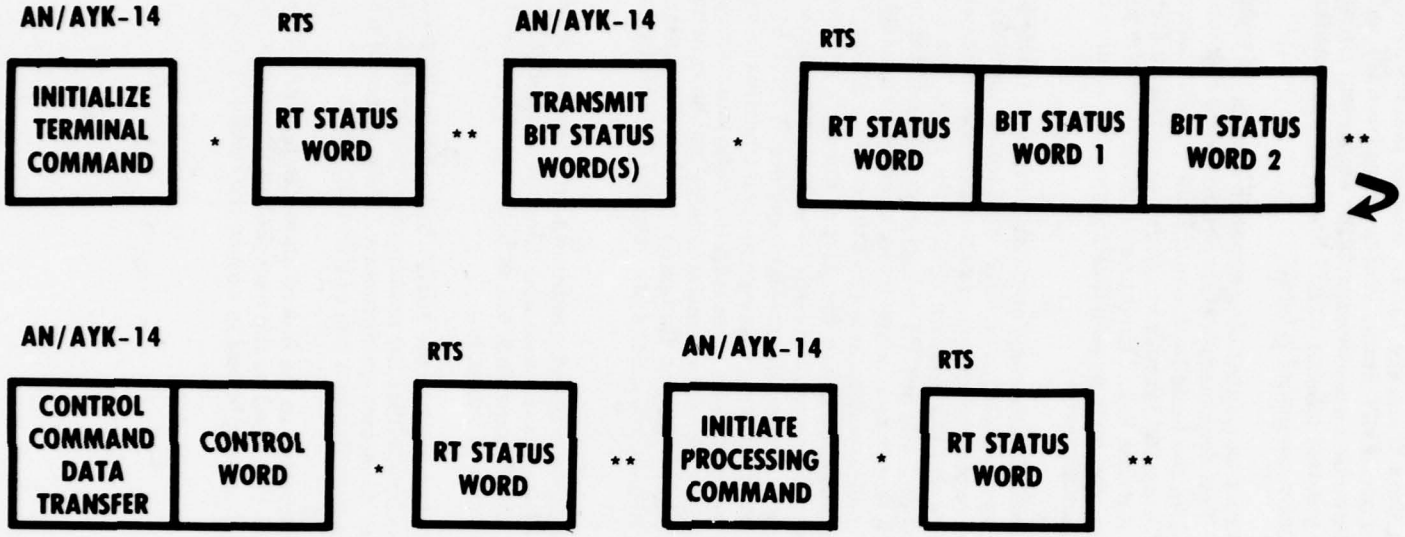



Figure 5 - Radio Terminal Set Initialization Sequence

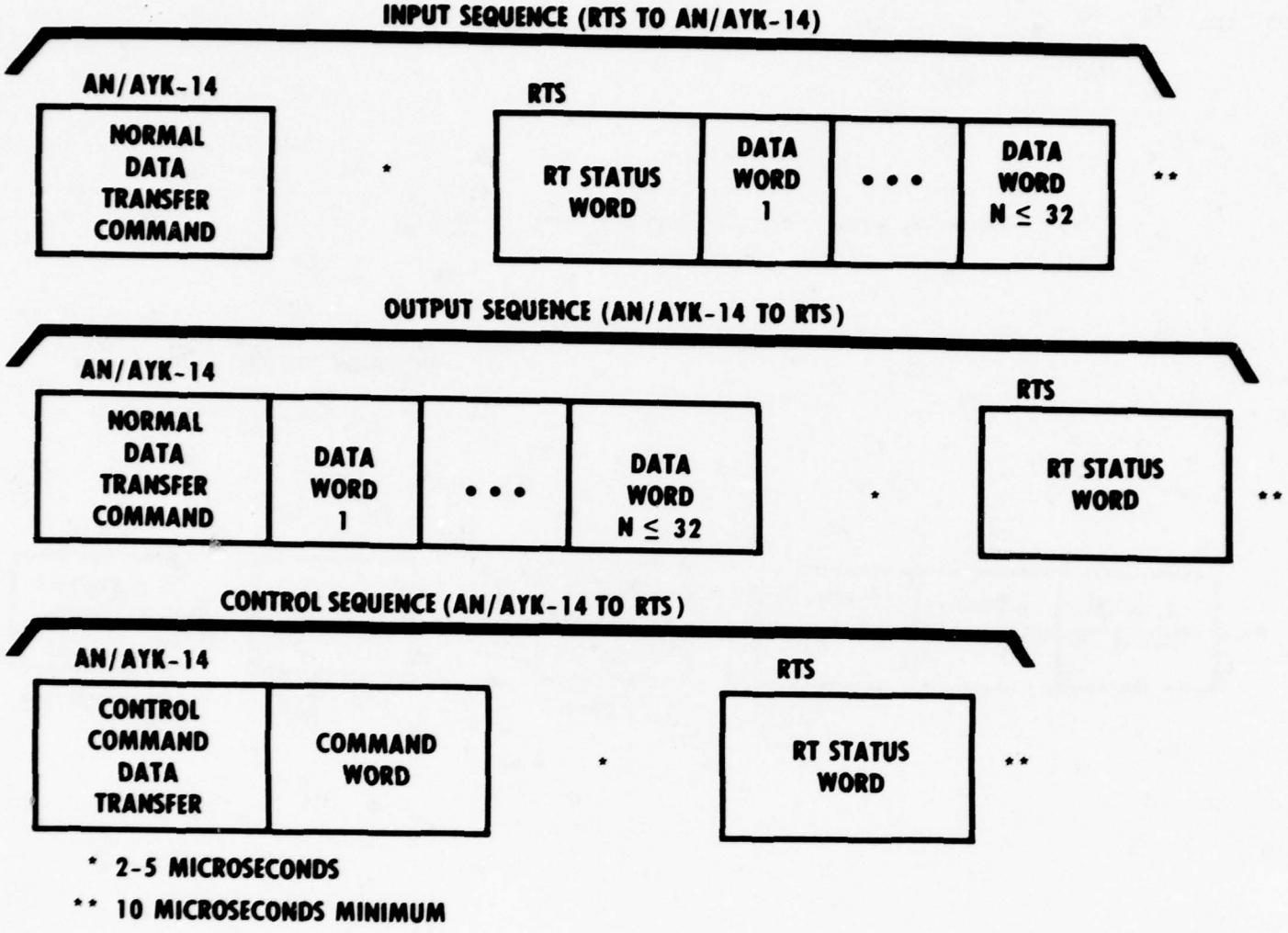



Figure 6 - Radio Terminal Set Data Transfer Sequences




* 2-5 MICROSECONDS

** 10 MICROSECONDS MINIMUM

Figure 7 - Radio Terminal Set Data Transfer Sequences

| 1553 A CHANNEL 0 | | | MAXIMUM TRANSFER | | | |
|---|---|---|---|---|---|---|
| | | | TERMINAL TO CONTROLLER | | CONTROLLER TO TERMINAL | |
| TERMINAL ADDRESS | EQUIPMENT CONNECTED | POLLING RATE (HZ) | DATA WORDS | PERIOD (MSEC) | DATA WORDS | PERIOD (MSEC) |
| 00001 | RTS (DATA LINK) | 25 | 32 | 40 | 32 | 40 |
| 00010 | EWS (ELECTRONIC WARFARE) | 25 | 32 | 40 | 32 | 40 |
| 00100 | SIGNAL DATA CONVERTER | 12½ | 2 | 1000 | 4 | 80 |
| 00111 | CSCG (COMM) | 5 | 29 | 1000 | 17 | 200 |
| 01000 | RNS (DOPPLER) | 5 | 3 | 200 | N/A | N/A |
| 01011 | CONVERTER MULTIPLEXER | 25 | 31 | 200 | 256 | 40 |
| 01110 | NSIU (NAV) | 5 | 2 | 1000 | 6 | 200 |
| 10000 | ATO CONTROL INDICATOR (KEYSET) | 5 | 2 | 200 | 10 | 200 |
| 11001 | ARMAMENT SDC (ORDNANCE) | 12½ | 2 | 80 | 1 | 80 |
| 11100 | SD CONTROL INDICATOR | 5 | 2 | 200 | 10 | 200 |

* AS REQUIRED

1553 CHANNEL 1 (BUS A OR BUS B)

| | | | | | | |
|---|---|---|---|---|---|---|
| 11111 | MAGNETC TAPE MEMORY | N/A | 2000 | 1000 | 2000 | 1000 |

Figure 8 - Lamps Mk-III Bus Controller/Remote Terminal Data Rates

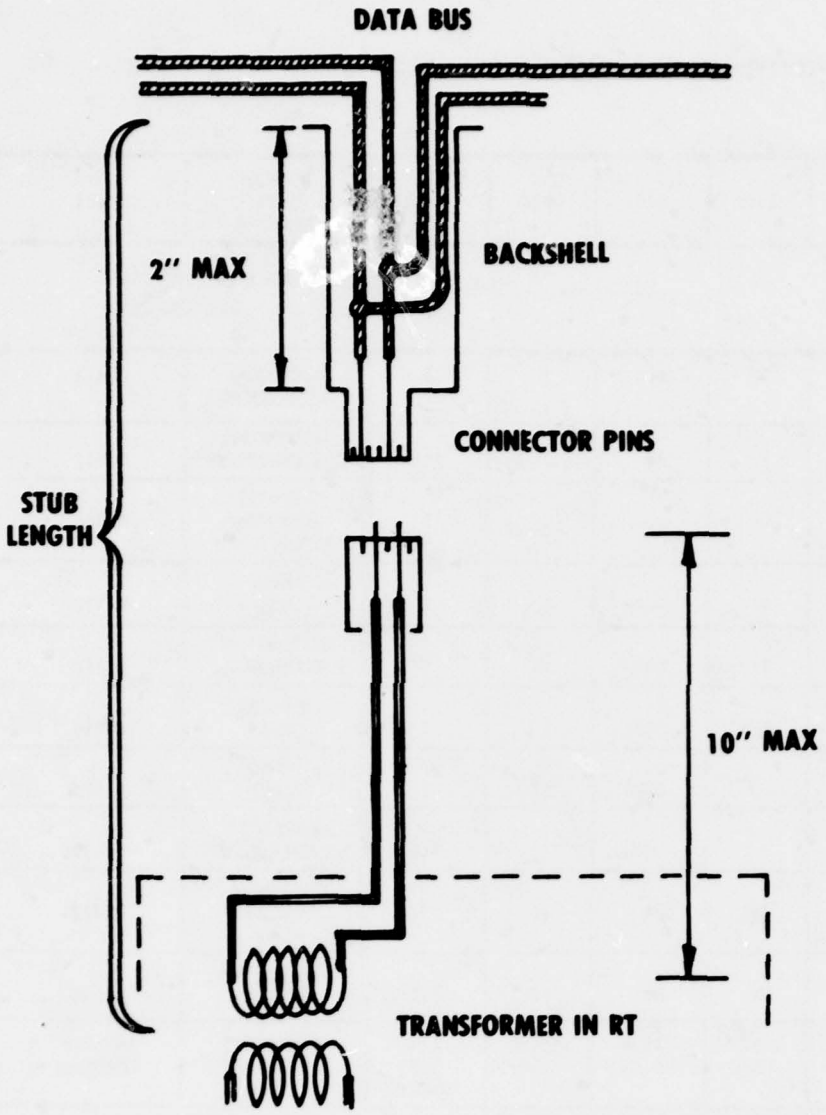



Figure 9 - Lamps Mk-III Stub to Bus Connection Method

## CONCLUSION

It is early in the LAMPS MK-III full scale development phase, and therefore it is difficult to draw firm conclusions. However, it is expected that several advantages will accrue to LAMPS because of the implementation of a 1553A data bus.

The system, the software, and many of the subsystems and subsystem elements are all simultaneously in development. The System prime contractor has been able to simulate all of the data bus interfaces, and because of this, is able to develop and test operational software before the actual equipment is available.

LAMPS MK-III is a complex weapon system, and system integration is expected to be a major effort. The prime contractor has devised a bus monitor which should facilitate the analysis and localization of integration problems.

The reliability of data transmission within the airborne system is expected to be higher than that demonstrated in earlier developmental configurations of LAMPS MK-III.

Finally, in the later phases of the system life cycle, replacement or addition of subsystems should be far easier and far less costly because of the decision to employ a 1553A data bus in the LAMPS MK-III.

# FRENCH UTILIZATION OF THE DIGITAL MULTIPLEX BUS

Bernard VANDECASTEELE

Ministry of Defence, Service Technique Aéronautique
France

Abstract : After a brief history of the studies which have been carried out till now in France about data buses, this paper describes the main features of the most used one. Then the development philosophy followed by French authorities is given and finally the past and current developments of modules, tools or equipments are shown.

## Historical backgrcund

The first serial digital bus was experimented in France in 1965, but the very beginning of the development of 1 MHZ digital multiplex buses took place in 1970, for use aboard strategic missiles. Since 1973, several studies for airborne applications have been conducted, some of them only as experimental studies, but some others as full development for major systems. One of the first studies was GINA (Gestion des Informations Numériques Aéroportées or Airborne Digital Informations Management) It will be presented to morrow in detail by M . QUIDET. This bus had a good success in France, since it has be chosen for several aircraft : Mirage 2000, the French new fighter ; Atlantic Nouvelle Génération, a new version of an existing maritime patrol aircraft ; and several versions of the F1 fighter. But it has also been chosen by the French Navy. This bus has already become de facto a military standard.

PRECEDING PAGE BLANK

Another study, merely oriented towards transport airplanes, has been developed and flight tested : it's SIGMA (Système d'Interconnexion Général Multiplexé pour Avions, or Airborne Multiplex General Interconnection System) which is now involved in an experimental program of avionic system's integration. The flight tests will be conducted on a Caravelle with a pretty name : ALIS, for Avion Laboratoire pour l'Intégration des Systèmes or Experimental Aircraft for Systems Integration.

Several other studies are now being conducted upon power management systems (E Mux) or for intercomm sub systems.

Although these systems present several differences, they all must adher to the Recommendations of the Technical Integration Committee (Comité Technique Intégration). This committee is a working group representing the services, pilots, manufacturers and equipment users (Air Force Staff and aircraft companies) created to collect views and experiences within the field of system integration, in order to define, if possible, French standards or to participate in the formulation of international standards. Since multiplex data bus techniques are one of the aids to system integration, they were among the first concerns of the Technical Integration Committee. Their recommendations, as well as the specifications of the GINA bus, have already been published in the AGARD Advisory Report N° 90 : A Study of Standardization Methods for Digital Guidance and Control Systems. I shall only give you here the main features of the GINA bus.

For more details, please refer to M . QUIDET'S paper.

## GINA main features

GINA is a 1 M bits/s digital multiplex bus, the general organization of which is shown on slide N° 1. You can see that each bus has two lines : LP (Procedure line) and LD (Data Line). Only the MOBs (Bus Monitor a Bus Controller) can speak on the procedure line, to give orders to the equipment. Every unit can hear and speak on the data line. This is done through a COS (Standard Coupler) what you call a Multiplex Terminal Interface Unit.

The second slide shows the principles of the exchange procedure. Each character has 10 bits : an 8 bits byte for data or command, 1 bit for validity and 1 bit for parity. During an exchange, the clock is given on the procedure line by the bus controller. So it is not necessary to have a clock in every unit, and this feature minimizes the risk of uncontrolled emission on the bus.

Slide N° 3 shows the structure of the standard coupler. The connection is made through transformers, for common mode rejection, to two receivers (one on each line) and one transmitter (only on the data line). The transmitter/receivers part can be duplicated if necessary for connection to a redundant bus.

Development philosophy

From the very beginning of the development, and apart from technical problems, a big question arose: what is to be specified? As the bus is a link between many equipment, there must be some development done by many manufacturers. In this way, the question was : what is standard, and what is specific ? Of course, the minimum is to have standard signals on the bus, i.e. on the wires. But we have also chosen to have a standard coupler, and therefore standard signals and procedures between the standard coupler and what we call the sub-system coupler, which is the specific interface between the unit and the bus. In all the programs which use a GINA bus, we have only one kind of standard coupler, and only one manufacturer, at least during the development phase (We'll have other sources for the production phase). Till now, this method has proven to be efficient, for several reasons :

- Development time : it's much more easy and fast, for an equipment manufacturer, to design the interface with a standard coupler (TTL level, parallel signals) rather than with a bus, and all the modifications or improvements that could be found interesting on the system's level can be implemented in all the equipment at a time.

.- Cost : we had only one development for this important card, instead of ten or twenty to do the same job ; in production, we will have an obvious advantage of a larger quantity of the same electronic module.

.- Technology homogeneity : I think that this point is as important as the cost aspect, because a failure in a poorly designed connection module can result in a failure of the whole avionic system.
So we have focused the studies on the safety of the standard coupler, so that in a system with about 20 units and a redundant bus controller, the safety of the all transmission system is made by that of the two controllers, the part of the couplers being negligible.Moreover, the connection design of all the equipment was aided and tested with the same tools, among which you find the Procedure Generator.

Past and current developments

The following slides show some modules, tools or equipment already developed for the bus.

First, the standard coupler, show on slide N° 4. Then, on the next slide, a bus controller which had been developed for the first flight tests of the bus. The most interesting thing can be seen at the rear side (slide N° 6) : we put a great emphasis on the technology of the wire and of the connector. The wire is a triply shielded pair, specially designed to have a high immunity, even for the Electro Magnetic Pulse, and to get the maximum of this cable, we developed a connector which insures good impedance and shielding continuity.

The next slide (N° 7) show a bus repeater, which permits the connection of a ground station on the aircraft bus at a distance of about 300 feet.

The two next slides (N° 8 and 9) shows the rear side of the bus repeater, with the bus connector, and it's internal cards.

On the next slide (N° 10) you can see the first version of the procedure generator, which is a development tool. Another version, controlled by a micro-processor, was also developed and can be used during the production phase.

The last slide (N° 11) represents an electronic bay which was made and used for the preliminary validation of the bus concepts.

Although there is a special session about systems integration, I would like to describe a few peculiar features of some systems now being developed in France.

In some applications, we have chosen to have a non-redundant bus, but a redundant bus management. This was possible because of the intrinsic redundancy of the GTNA bus, which has two lines . As the equipment can only hear on the procedure line, you have no risk of uncontrolled emission on this line, and a very little risk of short-circuit, so that you always can stop, through this line, a unit which disturbs the system through the data line.

Another characteristic of some of our systems is to be synchronous. For this purpose, the time, at a 50 pulses per second rate, is distributed on the bus to every unit. This clock can be used as an interrupt, to synchronize the data sampling and the program. I think that, when it's possible, a synchronous system is much more simple to design, because all the delays for sampling and transmission are constant and well known. Moreover, all the theoretical support, as the Z transform for example implicitly supposes that all the operations are synchronous, whith exactly the same frequency and known delays. Asynchronous systems must be used if there is no other possibility, because they are much more demanding, specially about the transmission specifications.

For the future trends, I just would like to say that many efforts are to be done to obtain low cost bus connection. For this purpose, we are now developing large scale integrated circuits, which will permit to save space, and especially, to cut down the costs.

# MULTIPLEXING IN ARMY TANK-AUTOMOTIVE VEHICLES

By

Jerry L. Perry
Systems Consultants, Inc.
Dayton Operating Center
Dayton, Ohio

Marquis W. Woody
U.S. Army Tank-Automotive
Research & Development Command
Warren, Michigan

## Abstract

The Army is now developing multiplex system hardware for application in certain of its ground vehicles. While the complexity of the electrical and electronic systems in such vehicles is considerably less than that encountered in aircraft, the results of a study* reported herein evidences that a significant potential payoff can be achieved through the use of multiplexing in Main Battle Tanks of the 55 to 60 ton class.

This paper summarizes the finding's of a seven-month study funded by the U.S. Army Tank-Automotive Research and Development Command (TARADCOM). The study constituted the first step in an Army program initiated at TARADCOM to develop Advanced Techniques for Electrical Power Management, Control, and Distribution Systems (ATEPS). The ATEPS program was initiated in the Fall of 1975. Subsequent to conclusion of the study reported in this paper, TARADCOM funded the development of ATEPS breadboard hardware, which conforms to MIL-STD-1553A, and has just recently been successfully demonstrated. More recently, TARADCOM has funded a study to determine the applicability of the ATEPS concept as well as the completed ATEPS hardware to other Army ground vehicles.

## Summary

Advanced electrical power management, control, and distribution techniques are required to meet the increasing power requirement impacts on combat vehicles. Since World War II, this power increase has been tenfold with major impacts of increased system complexity and reduced potential for system reliability. Vulnerability and survivability of wiring systems are, therefore, receiving increased Army attention.

---

* Perry, Jerry L., and Ziegler, Walter H., Jr., Requirements, Analysis and Conceptual Design of a Tank-Automotive Vehicle Information System, Laboratory Technical Report No. 12270, U. S. Army Tank-Automotive Research and Development Command, Warren, Michigan, 25 April 1977, Contract DAAE07-76-C-3264.

The initial ATEPS study centered around detailed analysis of the M60A2 which, at the time of the study, was the best documented vehicle in its class and was sufficiently complex to warrant its use as a baseline vehicle. Because conventional design techniques were employed in the M60A2, its electrical system was found to exhibit the following general properties:

- A large number of custom cables interconnecting the various subsystems and their dedicated controls and/or indicators;
- A large number of dedicated controls and indicators;
- A computer dedicated solely to the fire control function;
- A proliferation of relay logic to affect power control functions.

The most significant end result of the initial ATEPS study is the conceptual design of an information system for tanks. The information system utilizes advanced multiplexing, data processing, and integrated controls and display techniques. ATEPS thus integrates all electrical and electronic functions in the vehicle. The study demonstrates that the total Life Cycle Cost (LCC) of a Main Battle Tank employing such a system promises to be considerably less than that of present vehicles in its class. The information system proposed may be regarded as a general purpose utility which may be used by any subsystems that require the services it is capable of rendering.

Information Systems Description

An artist's concept of the information system as it might look when installed in a tank is illustrated in Figure 1. It consists of the following core elements:

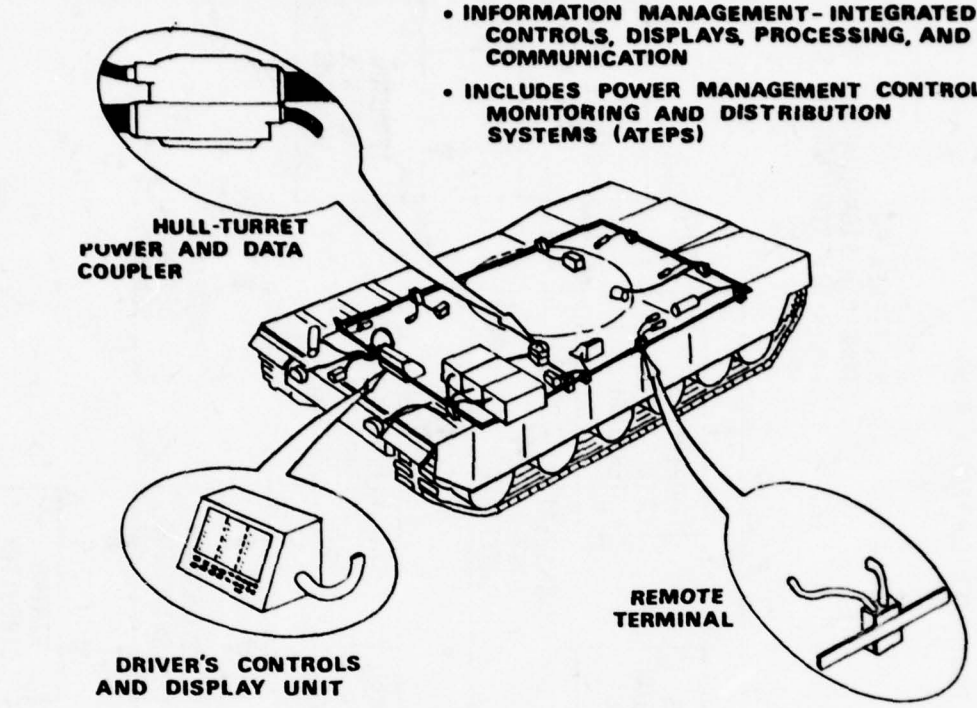
- An integrated power and data bus (flat cable)
- A number of compact Remote Terminals (RTs) which serve as interface devices between the bus and the subsystems it serves
- A Central Computer Complex (CCC) referred to in the figure as a dually redundant computer
- Four identical Crew Station Terminals (CSTs) which perform the integrated control and display functions.

It is a command/response time-division multiplex system under the overall control of a stored-program, general purpose digital computer. Key elements of the Central Computer Complex are the following: a 16 bit Central Processor Unit (CPU) working in conjunction with 32 thousand words of semi-conductor random access memory; a Bus Monitor and Controller (BMAC) working under the direct control of the CPU, a system control panel, system executive and applications software; a mass storage device; and miscellaneous power supplies, inverters, and special purpose system hardware. The flat cable Multiplex Data Bus distributes commands, data, and power throughout the vehicle. Figure 2 shows a top level block diagram of the system. It provides the following services:

- Electrical power distribution
- Computer-aided electrical power management and control
- Intercommunication between subsystems of subsystem sensor or control data via a multiplex data bus

## TANK AUTOMOTIVE VEHICLE INFORMATION SYSTEM

- INFORMATION MANAGEMENT - INTEGRATED CONTROLS, DISPLAYS, PROCESSING, AND COMMUNICATION
- INCLUDES POWER MANAGEMENT CONTROL, MONITORING AND DISTRIBUTION SYSTEMS (ATEPS)



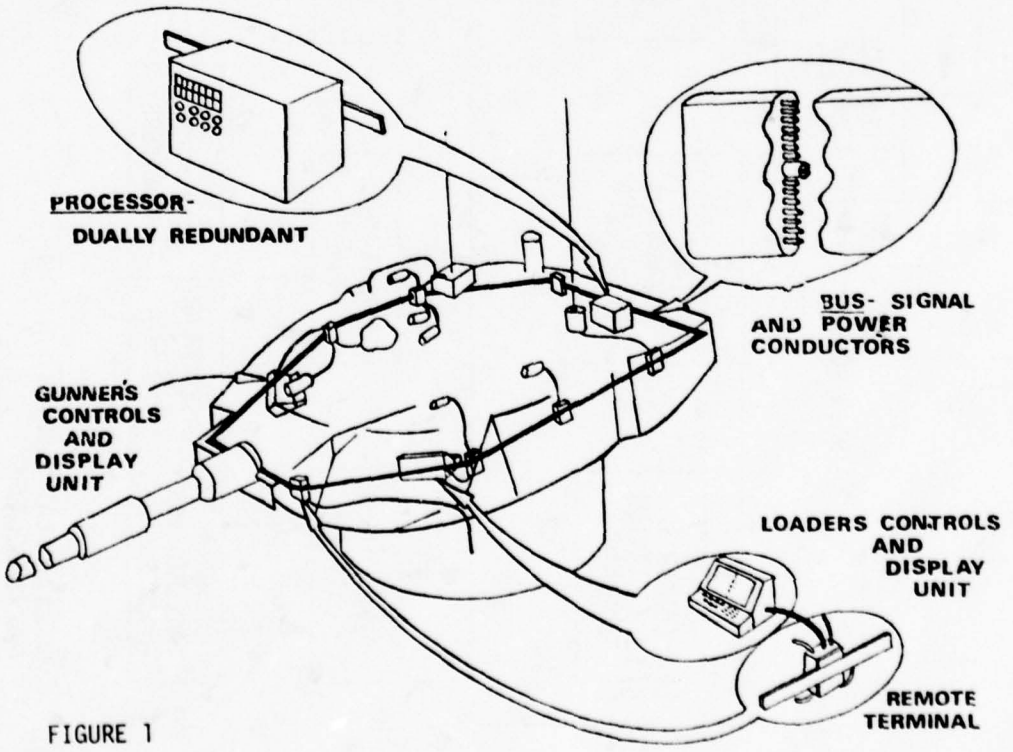




FIGURE 1

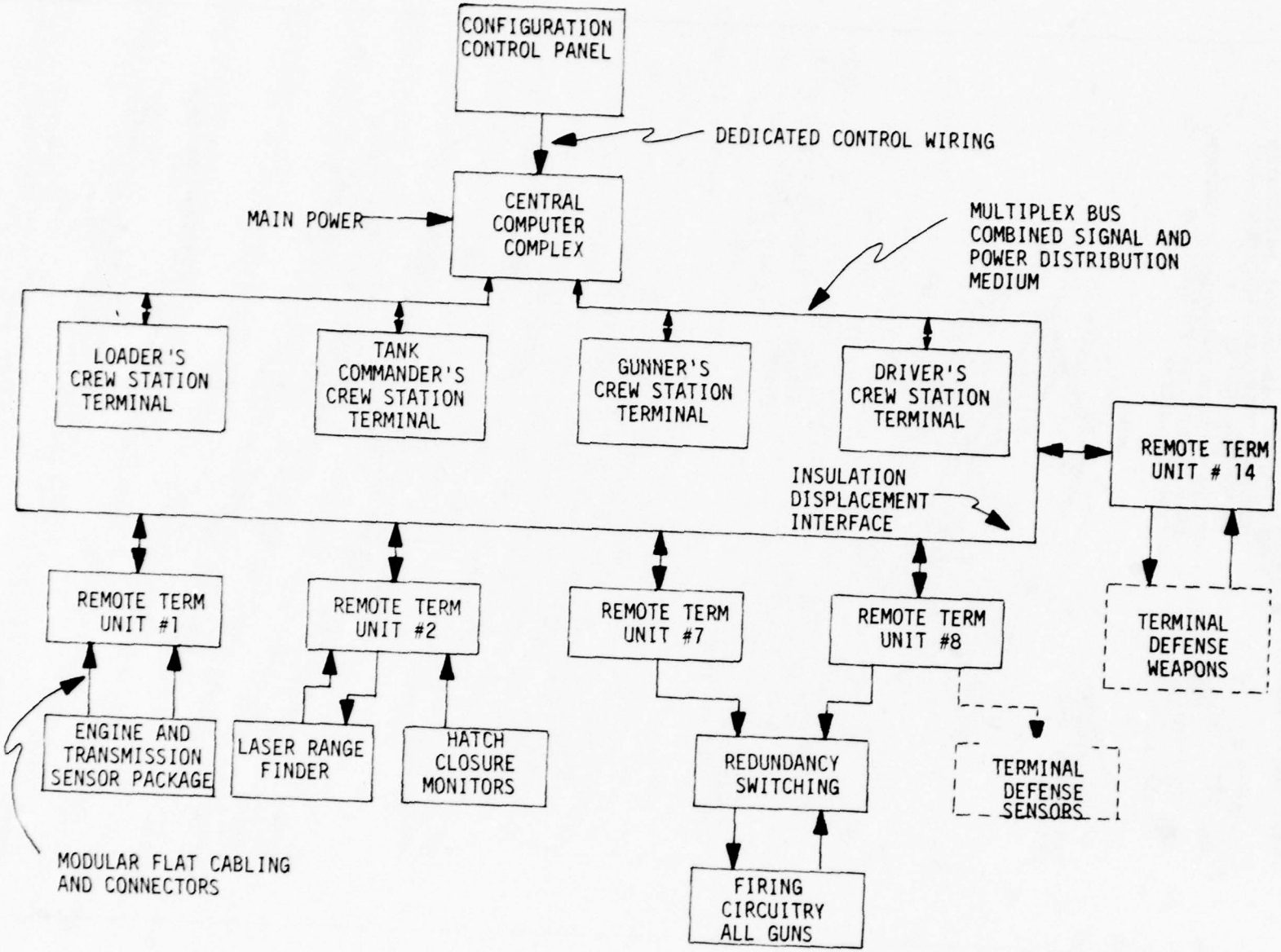



Figure 2 CONCEPTUAL SYSTEM BLOCK DIAGRAM

- Intercommunication between any crew member and the outside world via radio sets, etc.
- Data processing (such as fire control calculations)
- Control of subsystems via identical integrated multifunction manual input devices
- Display of measured or computed data via identical integrated multi-purpose display devices
- Malfunction detection and reporting
- Failure diagnosis
- Crew advisory service (such as display of check lists)

Its basic functions include the following: encoding of manual control inputs for selection and activation of processing operations, display of all information required to operate the vehicle, distribution of all command, control, display, and status information throughout the vehicle; automatic control; automatic status monitoring; and critical reconfiguration operations such as controlled load shedding (to provide graceful degradation). These basic functions are augmented (for purposes of establishing the system as a viable candidate for deployment) to include an expandable array of data processing, monitoring, and logging functions, additional peripheral subsystem interface growth capability, and survivability features that improve the useful product life. Principle features of the system are as follows:

- Continues to operate with no degradation of performance even if the power/data bus cable is severed.
- Redundant RTs can be employed selectively (but only where necessary) to serve failure critical subsystems and provide *graceful degradation*;
- Changes to the system or subsystem do not necessitate rewiring and/or overhaul of the vehicle;
- RTs may be incorporated inside the subsystems, thus simplifying the wiring from the bus to the subsystems it serves;
- Stand-alone RTs such as those shown in the figure may serve several subsystems in cases where it is not cost-effective to incorporate RTs directly inside the subsystems;
- No single failure in the CCC or any of the RTs will prevent the rest of the system from operating;
- The system monitors its own health, and automatically takes appropriate action should any of its elements malfunction;
- A power failure will not cause undesired output signals from the system to the subsystems it serves;
- The number of hull-to-turret slip rings is reduced to 14 including two spares (the M60A2 requires 52);
- Production connection of a subsystem or RT connector to the bus is a simple crimping process which greatly reduces wiring cost;
- The CCC and RTs are protected against NEMP (Nuclear Electro Magnetic Pulse Phenomena - the high voltage transients associated with nuclear explosions) pickup from the multiplex data bus;
- A novel interference cancellation technique suppresses EMI pickup by the multiplex data bus signal lines from the adjacent power lines in the same cable;

- Error detection and correction techniques permit system operation in the presence of any noise which fails to be rejected by the cancellation technique;
- The system reduces the number of independent switch position functions required by all 4 members of the M60A2 crew from 316 to 80;
- It reduces the required number of independent control and display equipments in the M60A2 from 12 to 4;
- It automatically calls the operator's attention to his display, when required, by means of an audio cue signal transmitted over the crew intercommunication system;
- It permits all electrical and electronic subsystem repairs to be effected without the need for depot overhaul facilities (and the resultant loss of vehicle availability for as much as a full year);
- The power and data bus will last longer than the service life of the vehicle;
- It can provide high-quality digital speech intercommunication between crew members or between any crew member and the radio sets;
- Its Crew Station Terminals (CSTs) are identical and interchangeable, thus reducing logistical and maintenance costs;
- The crew can remove and/or interchange CSTs in case of battle damage or CST malfunctions due to other causes (this is effectively a zero-cost, immediate-access spare capability);
- Each crew member's display can be displayed at any other crew station terminal;
- The crew may easily replace malfunctioning RTs with a limited number of spares carried on board the vehicle;
- The system can replace relay logic with programmable computer software logic, thus eliminating a considerable number of relays as well as the wiring between these relays;
- Because it is software programmable, system logic as well as system behavior in general may be modified without the need for hardware changes.

Figures 3 and 4 illustrate the physical characteristics of Remote Terminals and a Crew Station Terminal, respectively.

## Life Cycle Cost Investigation

Table 1 summarizes the results of the Life Cycle Cost (LCC) investigation conducted in parallel with the remainder of the study effort. It shows the expected impact of the system on each major identifiable LCC element for tanks. The most significant impact appears to be on the operating and support costs associated with depot overhaul as the ATEPS system will make it possible to effect almost all required repairs at the lowest maintenance level. Thus, the requirement for depot overhaul can be reduced in frequency if not eliminated altogether.

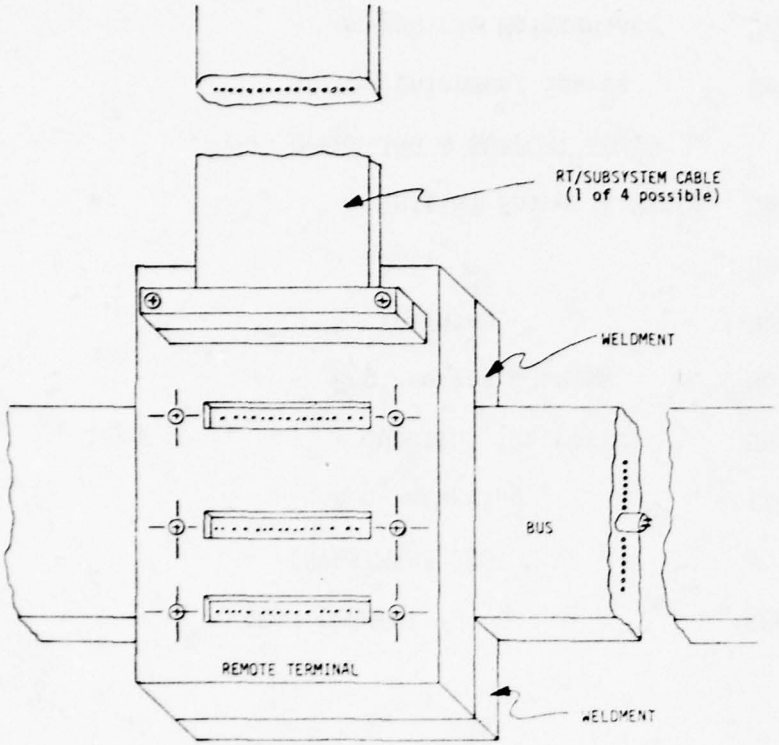
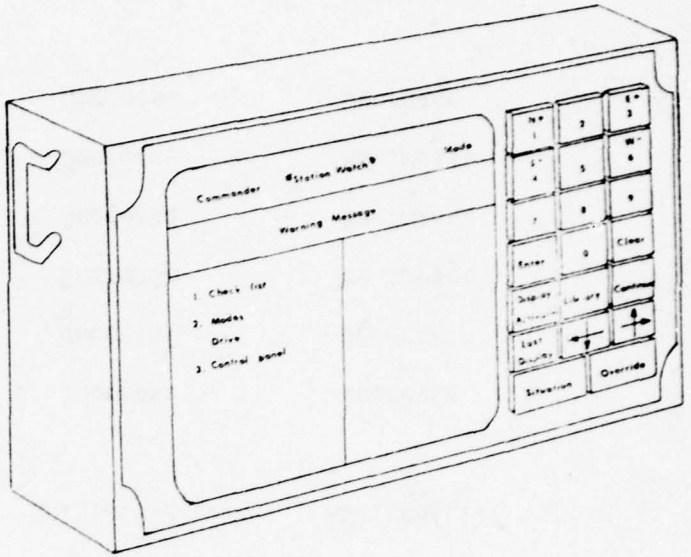



Figure 3 PHYSICAL DESCRIPTION OF REMOTE TERMINAL




SCREEN SIZE: 5 x 5.5 INCH

Figure 4 CREW STATION TERMINAL

TABLE 1 LCC IMPACT - LCC ELEMENTS

| COST ELEMENT | COST IMPACT OF IMPROVED RELIABILITY | MAINTAINABILITY | AVAILABILITY |
|---|---|---|---|
| - INVESTMENT COST | | | |
| - Non-recurring | Increase | Increase | Increase |
| - Production (Retrofit) | Uncertain | Uncertain | Uncertain |
| - Engineering Changes | No Impact | Decrease | No Impact |
| - Training | Decrease | Decrease | No Impact |
| - Data | No Impact | Decrease | No Impact |
| - Initial Spares & Parts | Decrease | Decrease | Decrease |
| - OPERATING & SUPPORT COSTS | | | |
| - Replenishment Spares | Decrease | Decrease | Decrease |
| - On-Vehicle Maintenance | Decrease | Decrease | No Impact |
| - Off-Vehicle Maintenance | Decrease | Decrease | No Impact |
| - Depot Overhaul | Decrease or Eliminate | Decrease or Eliminate | Decrease or Eliminate |
| - Modifications | No Impact | Decrease | Decrease |

## Methodology

Prior to designing the ATEPS system, the requirements of an M60A2 Main Battle Tank were rigorously analyzed. Its total electrical system was first broken down into subsystems as illustrated in Figure 5. All signals flowing between these subsystems were then listed, entered into a computer, and analyzed, using a computer program developed for the Army called MUXSYS. Table 2 is a sample computer printout of the computed output results. The output summary from MUXSYS indicates total amount of wire without multiplexing, required bus data rate with multiplexing, and the results of sorting signals by type. MUXSYS also computes the density of signal sources and destinations to assist in determining the best locations for remote terminals.

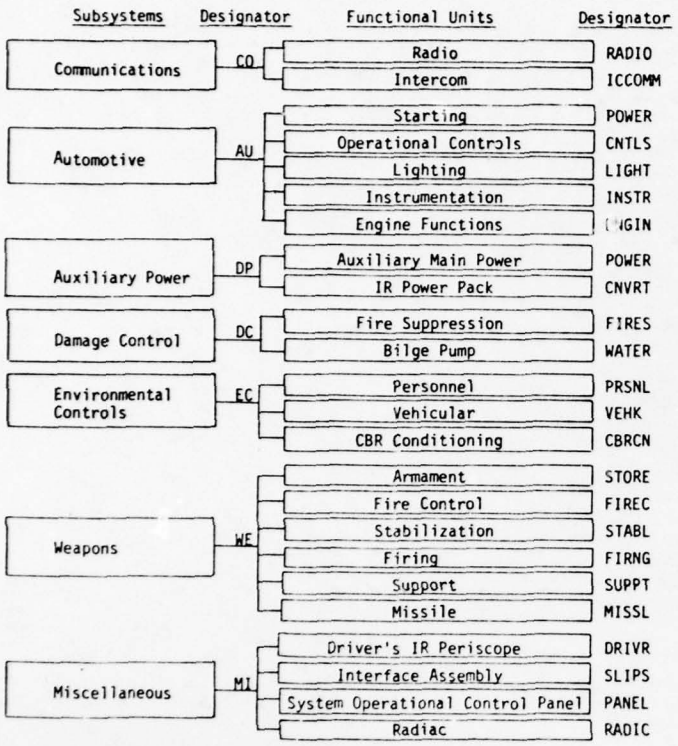



Figure 5 BASELINE ELECTRICAL SYSTEM BREAKDOWN

```
------ EQUIPMENT SUMMARY -----

NR. IN HULL     =     40
NR. IN TURRET   =     85
NR. IN CUPOLA   =     15
TOTAL NUMBER    =    140
TOTAL SYSTEMS   =     23

EQUIPMENT FUNCTIONS:

NR. CONTROL     =     11
NR. DISPLAY     =      4
NR. BOTH (C&D)  =     18
NR. NEITHER     =    107


----- RTU SUMMARY -----

NR. IN HULL     =      5
NR. IN TURRET   =      8
NR. IN CUPOLA   =      7
TOTAL NUMBER    =     20


----- SIGNAL SUMMARY -----

TOTAL NR. OF SIGNALS =         757
TOTAL LEAD LENGTH    =     26088.0 INCHES.
TOTAL INFO BANDWIDTH =       53444 BITS/SEC.
```

| SIGNAL ATTRIBUTE | ATT. VALUE | TOTAL LEAD LENGTH | INFO. BANDW. | TOTAL NUMBER |
|---|---|---|---|---|
| BY SIGNAL TYPES: | | | | |
| | Analog | 852.0 | 256 | 32 |
| | Digital | 21405.5 | 5088 | 615 |
| | Synchro | 3830.5 | 48100 | 110 |
| BY SIGNAL RESOLUTION: | | | | |
| | 1 | 21771.5 | 5188 | 625 |
| | 8 | 852.0 | 256 | 32 |
| | 12 | 3464.5 | 48000 | 100 |
| BY SAMPLE RATES: | | | | |
| | 1 | 6234.5 | 374 | 150 |
| | 10 | 16389.0 | 5070 | 507 |
| | 40 | 3464.5 | 48000 | 100 |

Table 2 MULTIPLEX SIGNAL SUMMARY OUTPUT

MINISTRY OF DEFENSE PROGRAMS IN SUPPORT OF MIL-STD-1553 RELATED RESEARCH

by

Dr. A. A. Callaway
Royal Aircraft Establishment
United Kingdom

(This paper was not submitted)

ESG

# SOFTWARE APPROACH TO MULTI-COMPUTER ARCHITECTURE USING MIL-STD-1553 DATABUS

H. Niedzballa
H. Ernst

Dok.-Nr.:
Ausgabedatum: October 1978
Ausgabe-Nr.:

Kopie-Nr.:

Alle Rechte vorbehalten. Vervielfältigung nur mit unserer Genehmigung gestattet.

ESG
ESG Elektronik-System-Gesellschaft mbH
Vogelweideplatz 9
8000 München 80
Tel. (089) 92 16-1, Telex 522594



PRECEDING PAGE BLANK

# SOFTWARE APPROACH TO MULTICOMPUTER SYSTEM ARCHITECTURE USING MIL-STD-1553 DATABUS

## 1. INTRODUCTION

The assumption can be made, that the design of electronic systems for weapon systems, especially for flying weapon systems, will be based on a standard data bus.

In comparison with point-to-point connections this means a fundamental reduction of safety. Therefore in any case a redundant data bus structure is necessary. Not only safety aspects require high reliability of the data-transmission-system. No failure within an electronic system should cause a break of the entire mission.

Therefore, if any of the data bus components (controller, transmission line, RT) fails, reconfiguration has to be performed immediately to achieve minimum system performance reduction. Reconfiguration is not only a matter of hardware. Reconfiguration should be based on the general software concept of the system, if optimum performance during all mission phases is desired.

The same problems occur when considering the computing capability which is available. This is the reason for the trend to distributed processing. It is therefore very useful to see datatransmission and dataprocessing as a closer problem, which should be solved by means of a proper software design.

(1)

The reason to emphasize the software approach is that systems up to now have been designed as a combination of equipment in terms of hardware. Thereby it has always been assumed, that sometimes later somebody will take

care of providing a software solution as a salvator for all the problems. Exactly this attitude leads to all the difficulties well known in software development until today. But this false standpoint has chances to survive by the phenomenon of the microprocessors. Many people believe, that cheap chips allow the easy design of systems with distributed intelligence. They don't see, that microprocessors of today are representing the state of the art of softwaretechnology in the 60ies.

By using microprocessors, system development is not made easier and cheaper, it becomes more complicated, more difficult to understand and more expensive in software development.

(2)

We therefore want to show a way, which in general leads to a safe system development by starting up from a safe software design. Multicomputer, multi-bus-structures are included.

## 2. (3) GENERAL METHOD FOR SOFTWARE APPROACH

Although applicable for all kinds of data-busses, this method will be explained under the assumption, that MIL-STD-1553 bus will be used in any one of its known versions.

### 2.1 Interface definition

In any case the once selected bus-structure is the interface between the process, represented by the peripheral equipment, and the software complex as a virtual machine, which establishes the required computer capability.

The real machine, and thereby the hardware configuration becomes its shape during solving the software problem.

2.2 Selection of software tools

(4)

For this the selection of the best software development system is necessary, which must be available at the time of the anticipated start of program development. It must include

- Real time programming language
- Translation system (i.e. compiler or interpreter)
- Real time operating system
- Test System
- Special support programs

2.3 Derivation of system functions

With respect to the constraints for realization given by the software system, the primary system functions for fulfillment of the mission will be derived according to the user's mission requirements.

2.4 Data Structure analysis

(5)

Each of these functions includes elements of data processing and data transmission, which have to be defined before the design of the computing system. That design is based on the time-quantities-budget for data manipulation and the computer capability required. To elaborate these requirements a very detailed study of the data structures of given equipments is necessary.

2.5 (6) Design of the virtual machine

After all this the virtual machine can be structured according to these data processing elements of the system functions. The criteria are:

- contingency of the elements, i.e. communication of tasks
- timing, i.e. synchronization of tasks
- complexity, i.e. number and size of tasks
- reconfigurability according to the requirements of the mission phases, i.e. generation of independent data processing modules as far as possible, which allow an effective failure management, and can be loaded dynamically.

2.6 (7) Real machine synthesis

The structure derived for the virtual machine can now be transfered to the real computer architecture. Number and arrangement of the computer elements (CP-, Memory-, I/O-, and communication units) follow immediately.

The structure of the virtual machine already implies a high degree of modularity, which is expressed by the existence of unified computing units in the sense of a computer family within the real machine. Thereby software compatibility and portability can be achieved.

Such computer design kit has big additional advantages by constructive simplicity, economy and, by this, problem-free logistics, because single as well as multi-computer complexes can be arranged from only a few different

elements.

If the requirement is accepted, that the task of any computer within the system can be taken over by any other, identical computers is the consequence. Though the bus-controller of the MIL-STD-1553 is primarily defined independently of a computer solution, and is a functional part of the data transmission system, a realization within the computing complex can have advantages with respect to failure behaviour and data exchange with computing programs.

## 2.7 Considerations on MIL-STD-1553

As already mentioned, this concept assumes MIL-STD-1553 in any of its variants. There are sometimes argumentations for changing that standard, e.g. improved interrupt handling and more flexibility of the transmission procedure.

There is no doubt, that a special modified MIL-STD-1553 bus could support a multicomputer concept even better than do existing ones. We don't like to exceed a standard in development if there is no essential need. What we would like to see, are extensions in the sense of upward-compatibility. Those are e.g. programmable bus-controller and RT, allowing useful improvements.

In aircrafts, solutions are favorized which allow very low couppled subsystems. For safety reasons flight control should be as far independent from navigation, weapon delivery, reconnaissance etc., as possible.

(8) A so-called inter-bus-coupler is proposed, an RT with passive connection to the bus-system of the FCS, which receives basically all the data transmitted. Data which are requested from the other subsystems will be stored in buffers and can be accessed via normal RT.

The critical FCS can be connected to the others either in the same way or via a normal RT or bus-controller.

To those meaningful extensions of MIL-STD-1553 we account the "DISCRETE BUS" presented here by VDO.

In principle the multicomputer system derived before does not need those extensions.

## 3. DEMONSTRATION EXAMPLE

### 3.1 Combat Aircraft Mission

We will now continue our presentation explaning the general method of software approach by an example.

How will such a computer complex look like in a weapon system, e.g. in a compbat aircraft.

First we present the typical tasks of an avionics system and its subsystems during the mission phases. The following phases are common to combat aircraft missions:

(9)

- Pre Flight
- Take Off
- Climb
- Rendezvous
- Cruise (outbound)
- Search and Loiter

- Combat
- Cruise (homebound)
- Landing
- Post flight

3.2 Avionics Subsystems

(10) The subsystems are:

| | |
|---|---|
| Weapon Delivery | WD |
| Navigation | NAV |
| Flight Control | FC |
| Communication | COM |
| Performance/Monitoring and Check Out | PM/CO |
| Electronic Counter-measures | ECM |
| Electronic Counter/ Counter Measures | ECCM |

3.3 Mission/Subsystem Matrix

(11) By introducing priority numbers a mission phases/Subsystem-Matrix can be established.

3.4 Software Approach and Design

3.4.1 Matrix Analysis

The matrix is showing the 6 subsystems which are the 6 main tasks having changing priorities. They are running under the control of an RT-OS still to be defined, which administrates inter-task-communication and synchroization. The virtual machine can also still not include a failure strategy, but first inputs are given by the priorities.

### 3.4.2 Software design requirements

(12)

On the software side a global address space has to be defined, which covers the whole system. The programming system to be employed must allow the designer to formulate tasks without respect to equipment structure of the computing system by language features.

This means, application of a higher level real time programming language with features as follwos:

- Description of real time operation and special distribution of the computing system including

  - o Timing and syschronization
  - o Administration of resources
  - o File Management
  - o Process I/O
  - o handling of process information (Bit-handling)
  - o Description of processor type,-location, -address range
  - o Allocation of programs to computers
  - o Allocation of I/O-Equipment and data of the technical system to variables of the problem-oriented program
  - o Description of program communication

- Fail Soft Handling

- Allocation of task to computer depending on the status of the computing system

- Description of alternative ways for process data acquisition of the computing system, the I/O system, the data transmission system (e.g. databus) and sensors/ effectors.

These criteria are met by the high level RT-language PEARL (Process and Experiment Automation Real-time Language) already to a high degree. First extensions including description of multicomputer structures are now developed and will be available for software development in a few years.

## 3.5 Real Machine synthesis

If those software tools are to be applied, the real machine can be structured now.

### 3.5.1 Description of the elements

(13) Considerations on cooperation and distribution have shown, that a complex of four identical computers connected by bus systems, will be reasonable.

The bus-systems can be divided into two groups:

- The computer complex internal, serial memory- and communications bus (S-bus for systems bus)

- The computer complex external process bus (P-bus) according MIL-STD-1553 "X".

The modular computer unit consists of the following components:

- CPU
- Local Program Memory
- Communications Memory
- Bus Controller for the access to additional memories within the multicomputer system
- I/O-controller for the access to sensors/effectors of the avionics system
- Fault detection logic.

The global memory attached to the S-bus is accessible for each computer and allows dynamic reloading, i.e. reconfiguration of software packages.

### 3.5.2 Description of Functions

The function of the computer complex is as follows:

In normal mode of operation each computer has to fulfill a given set of tasks, which is designed in order to achieve homogenuous load, not too much reconfiguration, minimum interdependence with other sets of tasks, and total system integrity.

In case of disturbance or failure of computer components or their related data transmission systems, the tasks will be newly assigned by a predetermined failure strategy.

The assignment follows the principle of "Dynamic Hierarchy", which means, that under all system states, each computer can be assigned to each task. One computer will always have system control, which alternates in case of a failure to another according to the failure strategy.

### 3.5.3 Description of Task Arrangement and Fail Soft Strategy

A possible task assignment during the mission phases could be performed as follows:

(14) A complete decision table must also cover fault information about any 1553 X bus-components (controller, transmission, RT). The table shows only the most important decision rules. Those concerning degraded subsystem performance must be added.

### 3.5.3 Operating system design aspects

Reconfiguration according fail soft strategy requires a specific organisation of the operating system:

- Local OS contains all functions for process programs and communication. A stand alone operation must be possible.

- Global OS contains in addition system start functions, decision table and monitoring functions.

(15) We leave these considerations now and will present a few
(16) pictures of structuring a virtual machine based on PEARL
(17) as design and programming language. We hope, these viewgraphs are self-explanatory now for the audience.

## 4. CONCLUSIONS

- Advantages

  o Very high probability of mission success

  o Low effort for changing SW as well caused by technological retrofit as by system modification

  o Standard design kit existing from few computing- and data transmission components (i.e. MIL-STD-1553X)

  o Applicable for very different weapon systems because of adaptability of the configurations to the required computing power (single/-multicomputer)

  o Generalized system design method to be transferred to all applications; hereby low effort in system-development in further projects

- o Very high objectivation of the system by high transparency, therefore no more critically dependant on persons

- o Projects today rated as highly sophisticated, become feasible by better transparency and overview.

- Disadvantages

  - o Looks like additional HW-Effort

  - o Successful fight for a standard is necessary

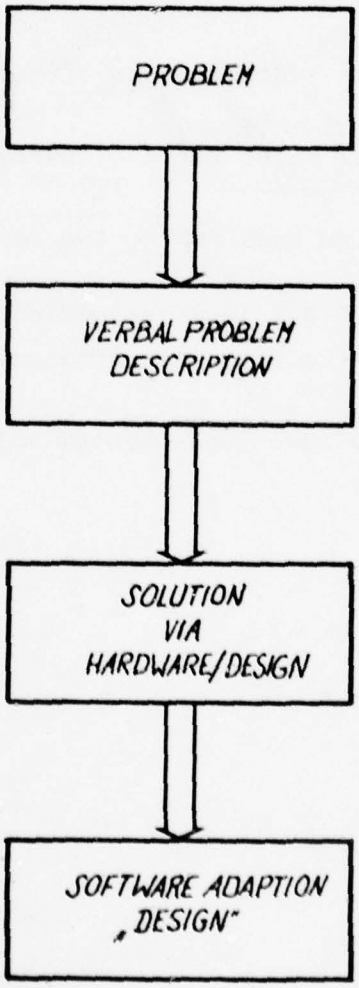
  - o System size must not be too small

  - o Eventually difficult specification of the reconfiguration or the fail soft strategy

  - o Initially additional system software development necessary.

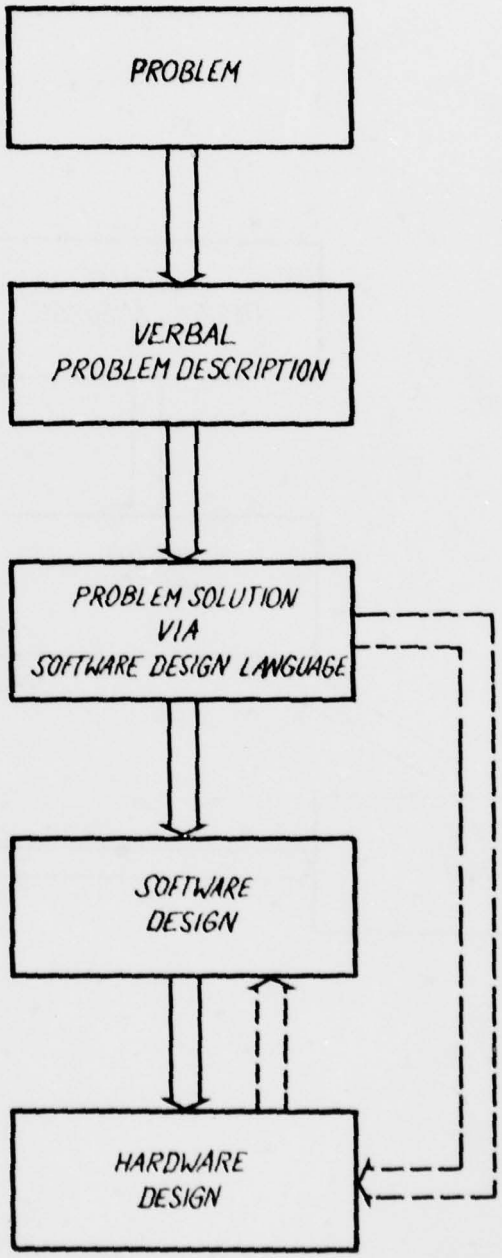
ESG

## MOST COMMON SYSTEM DESIGN APPROACH



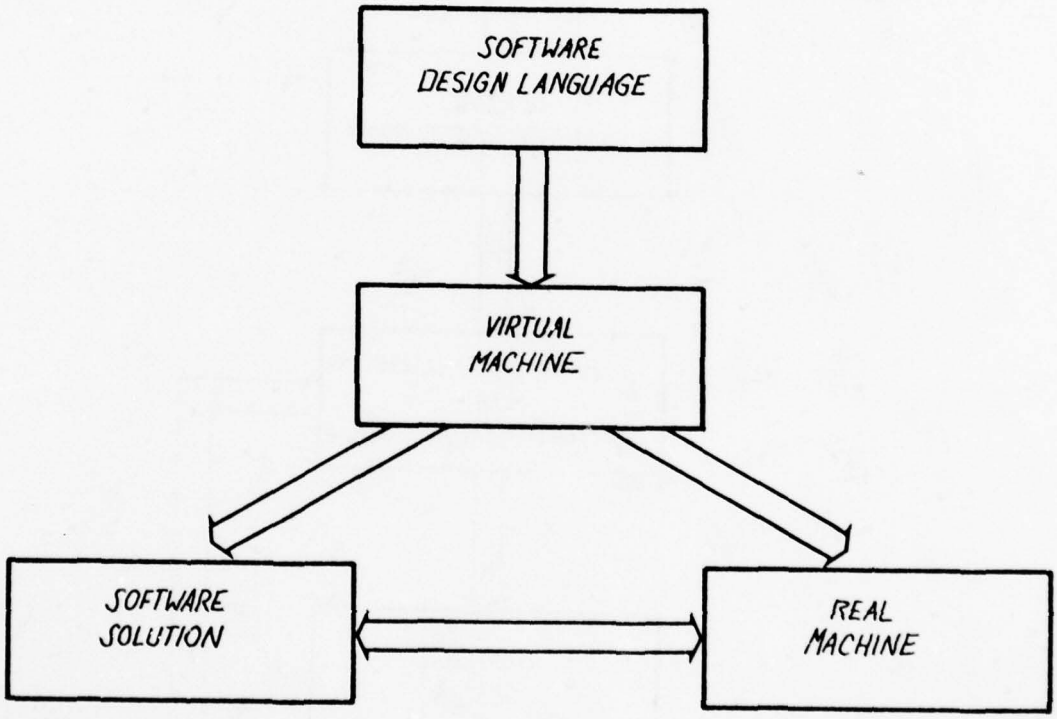
ESG

## PROPOSED SYSTEM DESIGN APPROACH



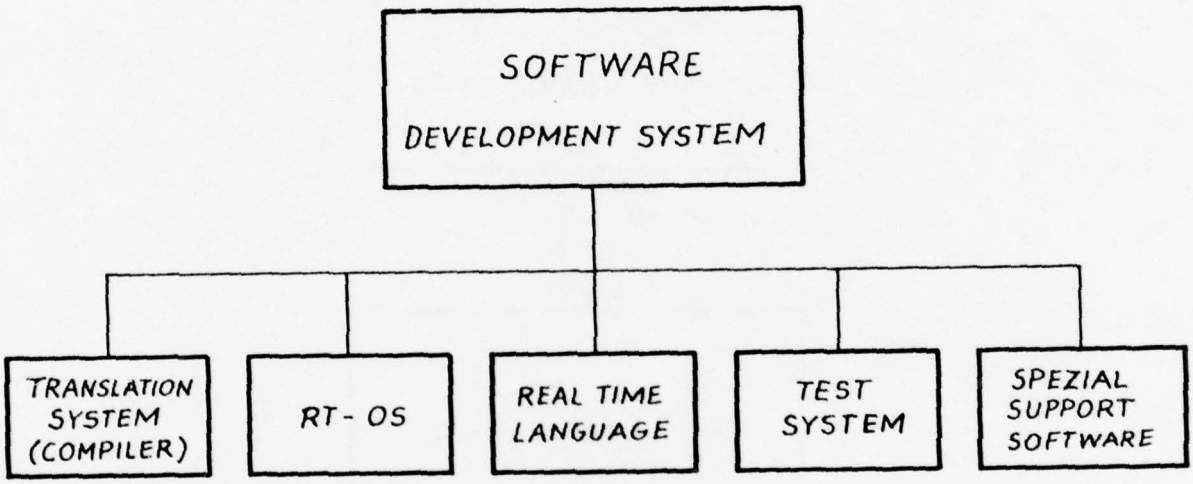
ESG

## DEVELOPMENT STEPS



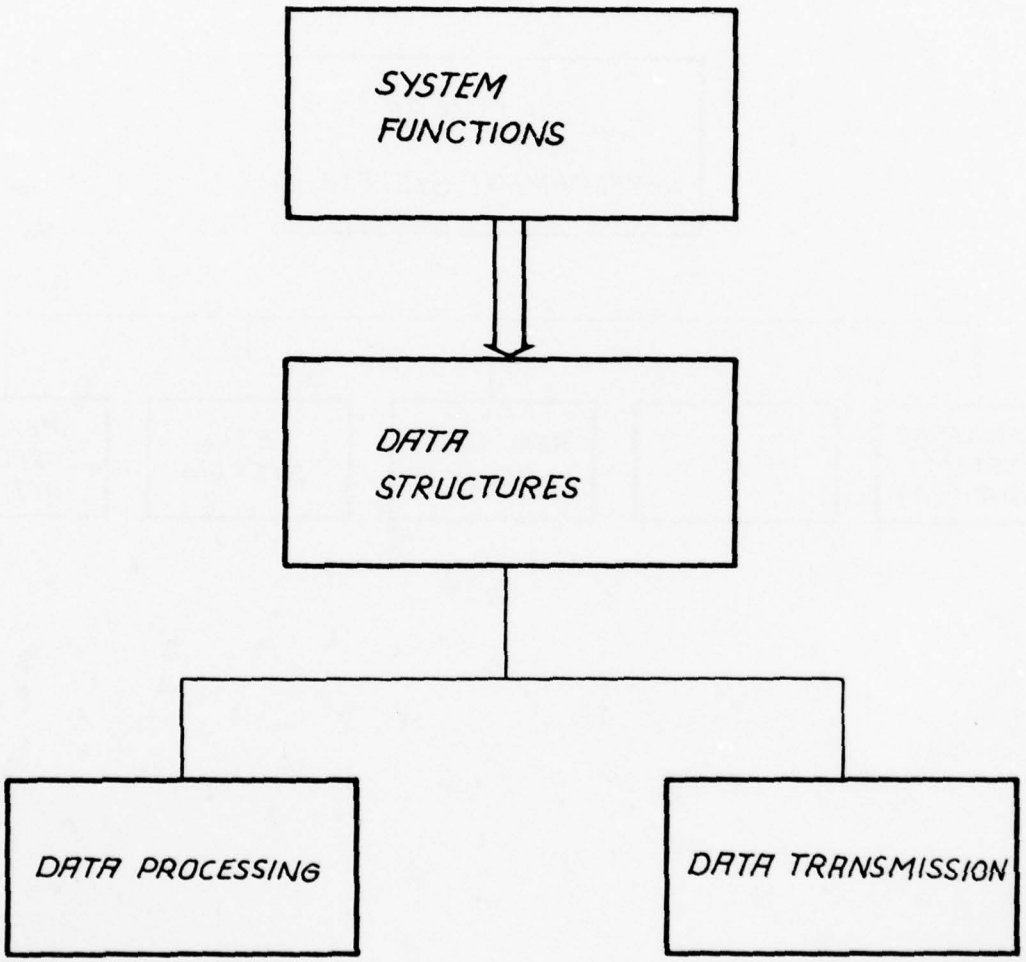
ESG

## COMPONENTS REQUIRED FOR SOFTWARE DEVELOPMENT



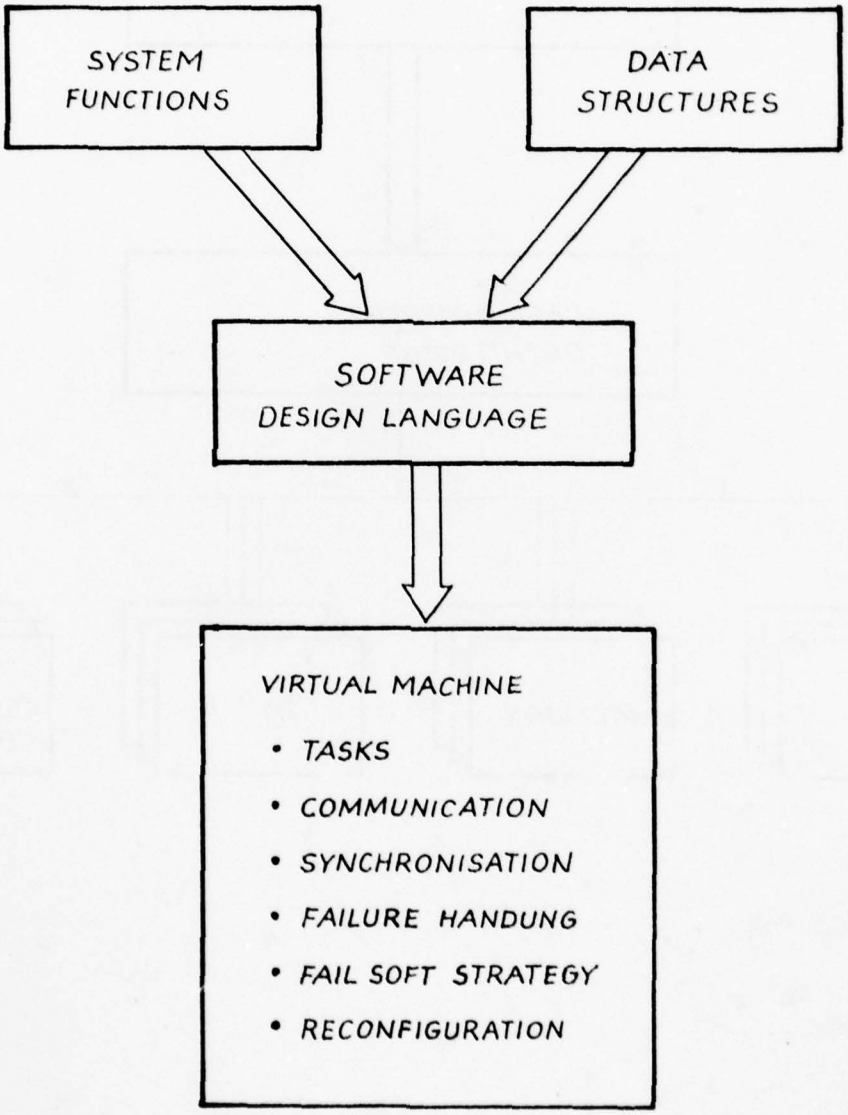
ESG

# PRIMARY DESIGN STEPS



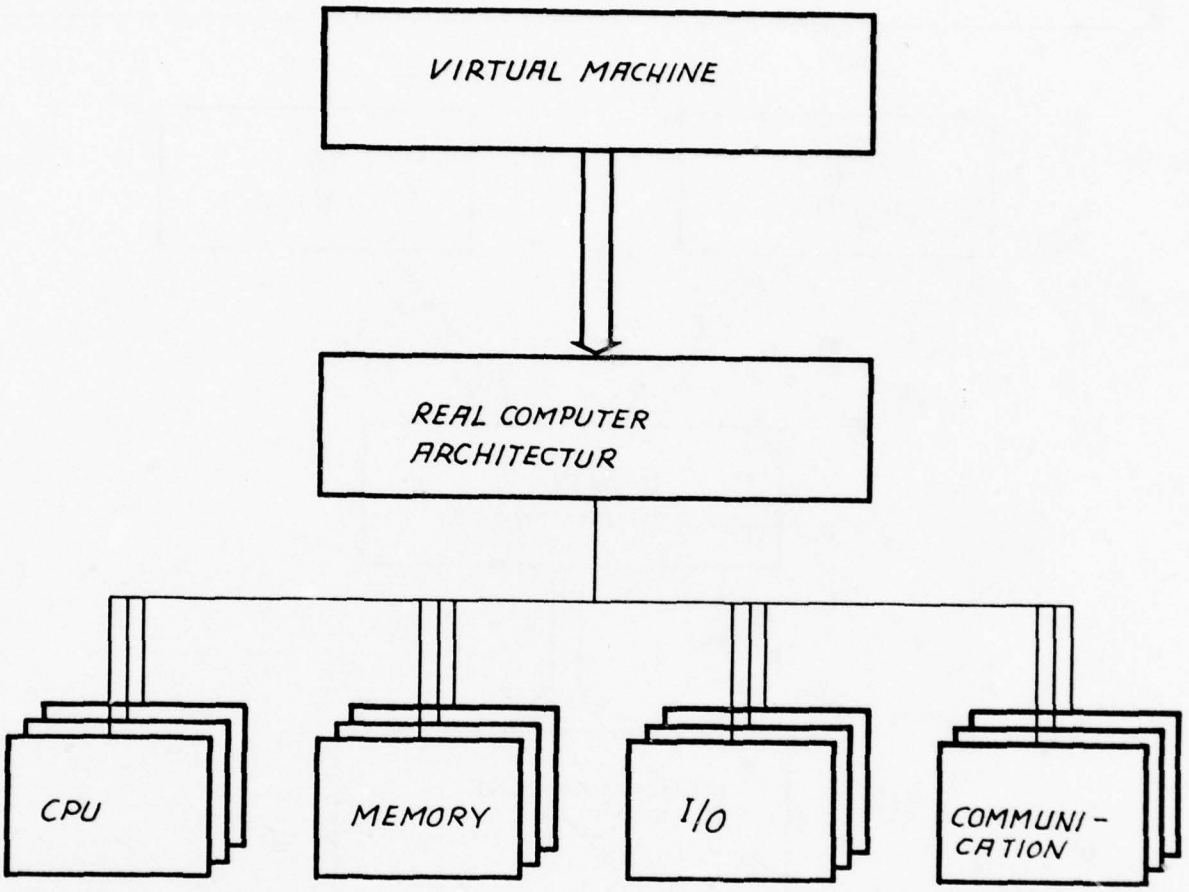
ESG

# DESIGN OF VIRTUAL MACHINE



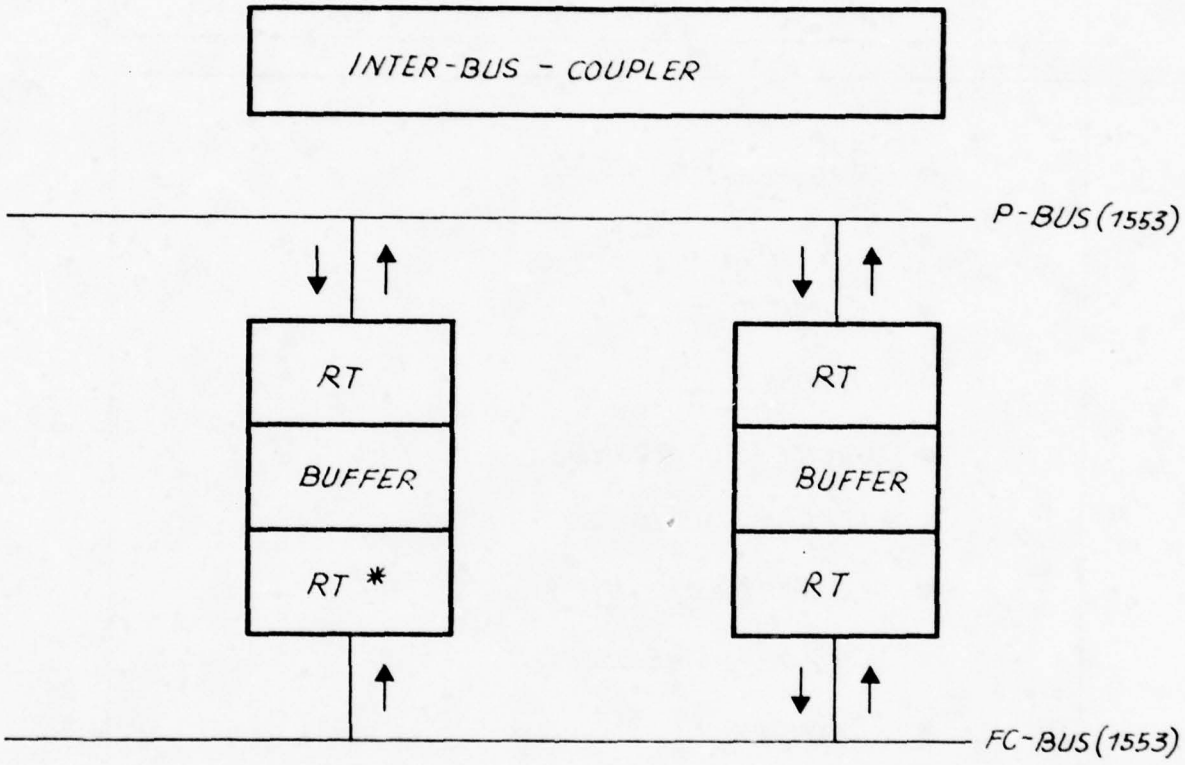
ESG

# REAL MACHINE SYNTHESIS



ESG

# PROPOSED EXTENSIONS OF MIL-STD 1553 BUS




* REMOTE TERMINAL WITH RECEIVE ONLY FUNCTION

ESG

## MISSION PHASES OF A COMBAT AIRCRAFT

- PRE FLIGHT
- TAKE OFF
- CLIMB
- RENDEZVOUS
- CRUISE (OUT BOUND)
- SEARCH AND LOITER
- COMBAT (A/A, A/E)
- CRUISE (HOME BOUND)
- LANDING
- POST FLIGHT

ESG

## AVIONICS SUBSYSTEMS

- WEAPON DELIVERY (WD)
- FLIGHT CONTROL (FC)
- NAVIGATION (NAV)
- COMMUNICATION (COM)
- PERFORMANCE MONITORING/ ON BORD CHECK OUT (PM/CO)
- ELECTRONIC COUNTERMEASURES (ECM/ECCM)
  ELECTRONIC COUNTER COUNTERMEASURES

ESG

# MISSION / SUBSYSTEM MATRIX

| PHASE \ TASK | WD | FC | NAV | COM | PM/CO | ECM/ECCM |
|---|---|---|---|---|---|---|
| PREFLIGHT | O | O | O | O | + | O |
| TAKE OFF | — | ++ | O | O | ++ | — |
| CLIMB | — | ++ | O | — | ++ | — |
| RENDEZVOUS | — | ++ | + | + | ++ | O |
| CRUISE (OB) | — | ++ | ++ | O | ++ | O |
| SEARCH/LOITER | O | ++ | + | O | ++ | + |
| COMBAT | ++ | ++ | O | — | ++ | ++ |
| CRUISE (HB) | — | ++ | ++ | O | ++ | O |
| LANDING | — | ++ | + | ++ | ++ | — |
| POST FLIGHT | O | O | O | O | + | O |

++ HIGHEST PRIORITY

\+ HIGH PRIORITY

O NECESSARY

— NOT NECESSARY

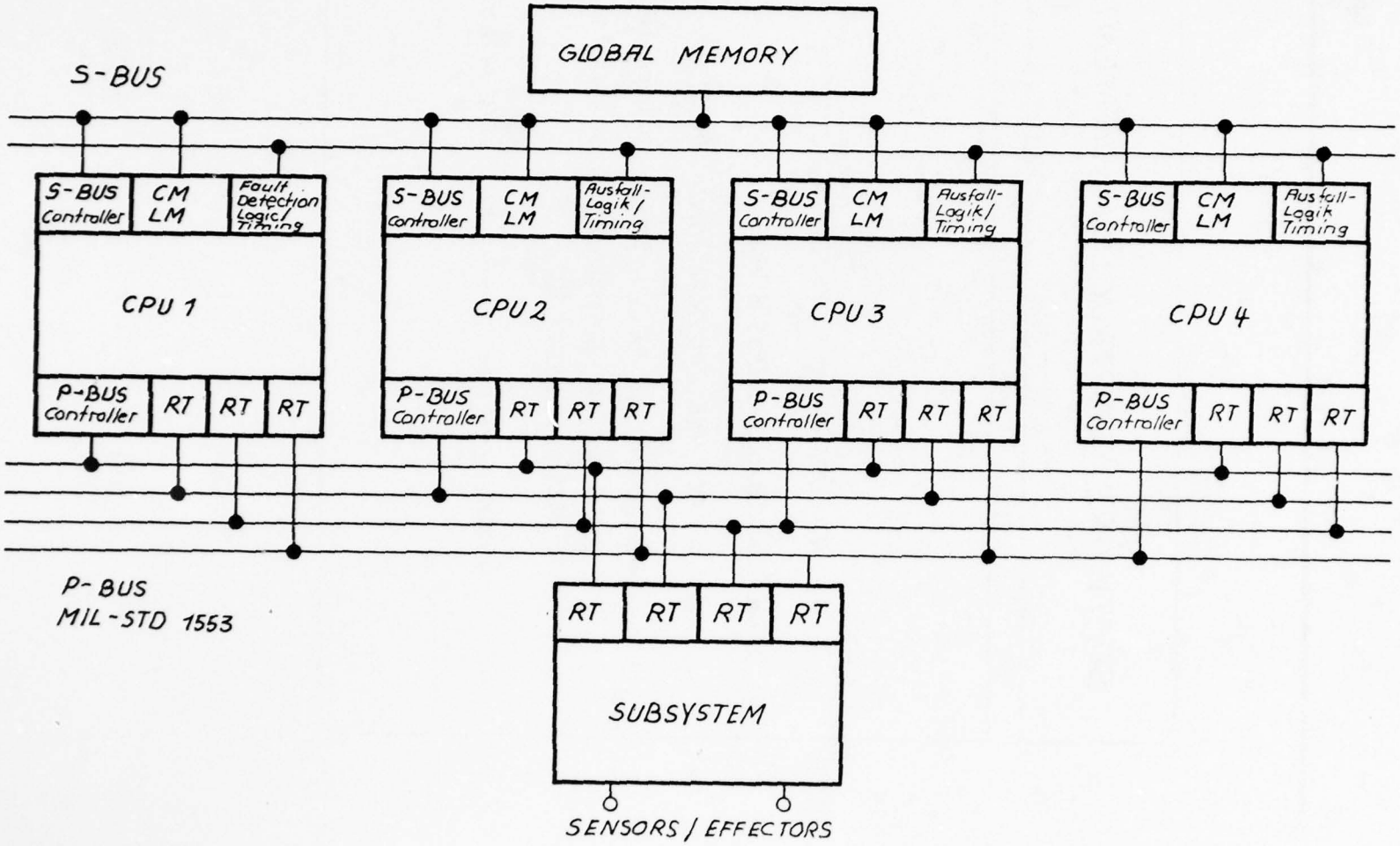
ESG

## SOFTWARE CONSTRUCTION ELEMENTS

- RT - OPERATING SYSTEM
- GLOBAL ADRESS SPACE
- REAL TIME PROGRAMMING LANGUAGE
  - DESCRIPTION OF REAL TIME OPERATION AND SPATIAL DISTRIBUTION OF THE COMPUTER SYSTEM
  - PROCEDURES FOR FAIL SAFE OPERATION

ESG

# 4 COMPUTER SYSTEM STRUCTURE



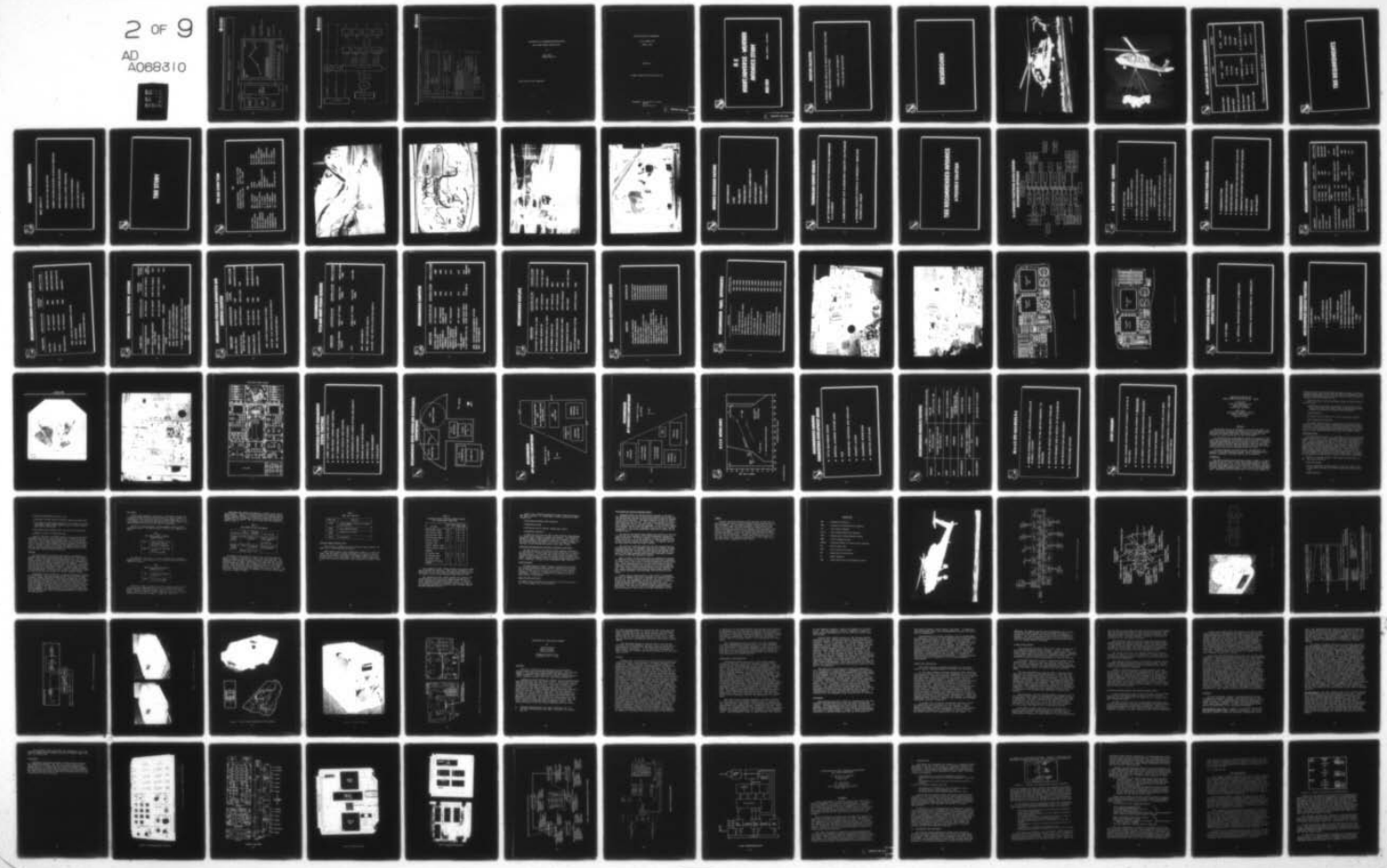
ESG

## SOME RULES OF THE DECISION TABLE FOR TAKE OFF

| | R1 | R2 | R3 | R4 | R5 | R6 | R7 |
|---|---|---|---|---|---|---|---|
| CPU 1 | T | F | T | T | T | F | F |
| CPU 2 | T | T | F | T | T | F | F |
| CPU 3 | T | T | T | F | T | T | F |
| CPU 4 | T | T | T | T | F | T | T |
| WD | — | — | — | — | — | — | — |
| FC | 1,2 | 2 | 1 | 1,2 | 1,2 | 3 | 4 |
| NAV | 3 | 3 | 3 | 2 | 2 | 3 | — |
| COM | 3 | 3 | 3 | 4 | 3 | 4 | — |
| ECM/ECCM | — | — | — | — | — | — | — |
| PM/CO | 4 | 4 | 4 | 4 | 3 | 4 | — |

AD-A068 310 AERONAUTICAL SYSTEMS DIV WRIGHT-PATTERSON AFB OH F/G 17/?
PROCEEDINGS OF THE AFSC MULTIPLEX DATA BUS CONFERENCE (2ND), HE--E
OCT 78 E C GANGL
UNCLASSIFIED ASD-TR-78-34-VOL-2 NL

2 OF 9
AD A068310



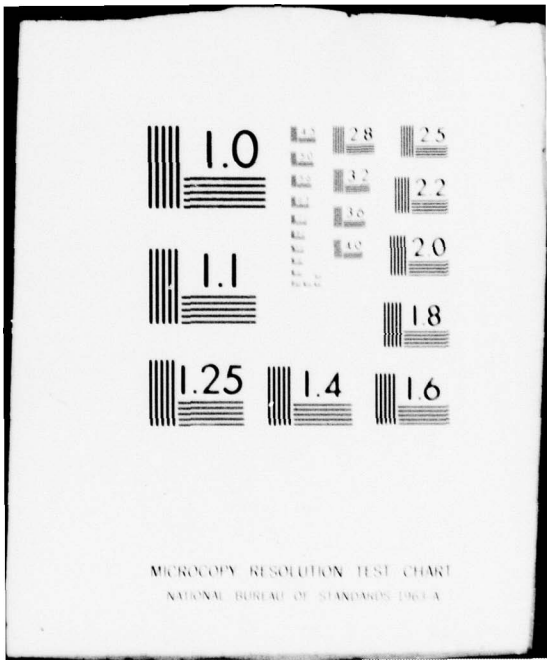
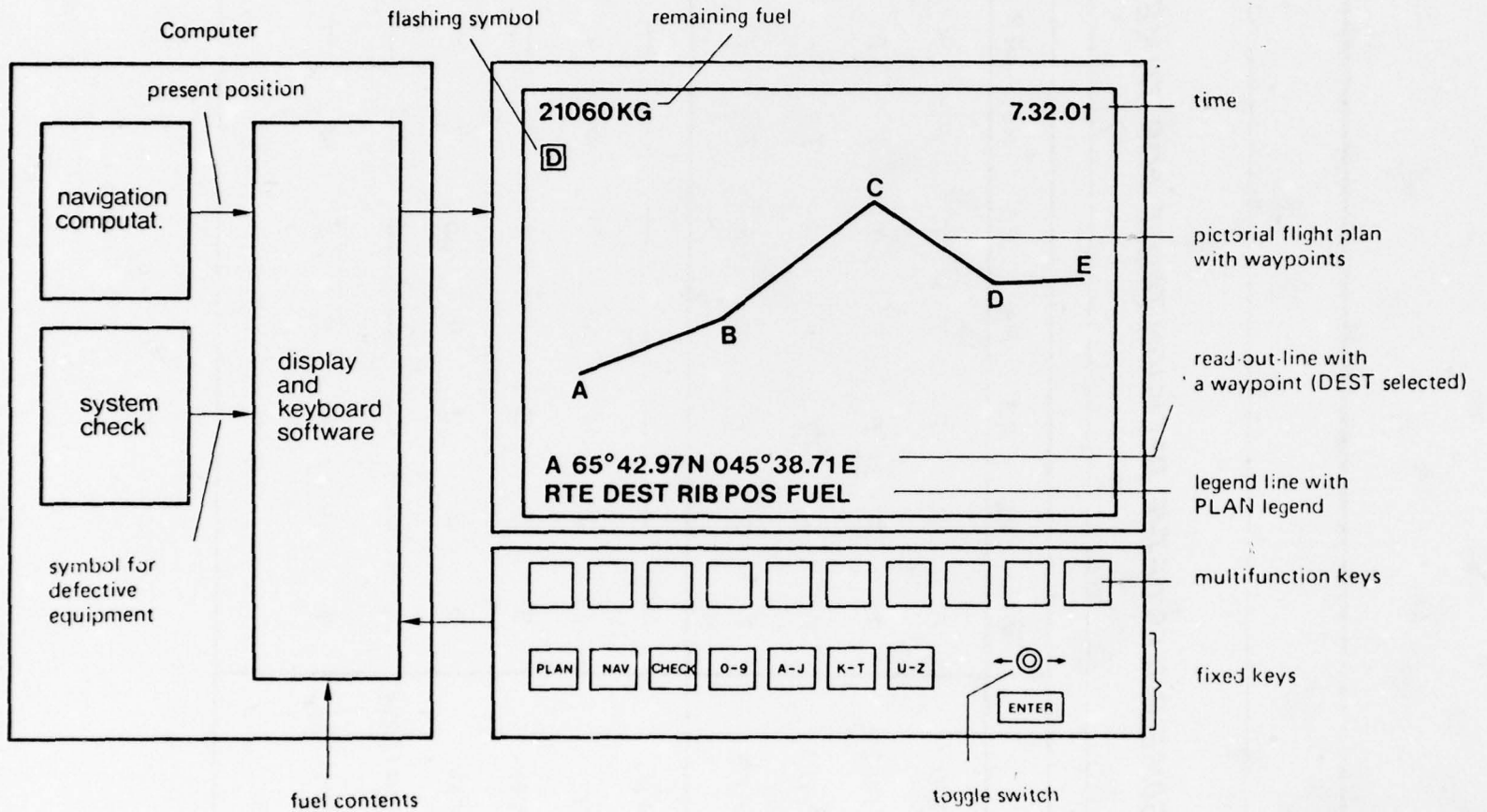

1.0
1.1
1.25
1.4
1.6
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.8
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

Figure A1.1: Environment of the display and keyboard software
(PLAN format selected)



ESG
16
CALL
CALL
CALL
CALL
FUEL
R&R
DEST
ROUTE
ACTIVATE
TERMINATE
REQUEST
RELEASE
additional formats
NAV
PLAN
ALPHANUMERIC
Tasks
CALL
REQUEST
RELEASE
ACTIVATE
TERMINATE
REQUEST
RELEASE
ACTIVATE
TERMINATE
REQUEST
RELEASE
ACTIVATE
TERMINATE
FLASH SYMBOL
TIMER
KEYSCAN and subroutines
CALL
TAKE (Input)
Keyboard
Display
SEND (Output)
DISPLAY ROUTINES
ACTIVATE
ACTIVATE with schedule
ACTIVATE
STARTER TASK

ESG

```
PEARL-AVIONIC-SUBSET  VERSION 0377/01   MODULE: MAIN                    DATUM 24.05.77   ZEIT 20:39:12   PAGE:    1


      1   MODULE(MAIN);
      2
      3   /*************************        MAIN        *************************/
      4   /*                                                                    */
      5   /*  THIS MODULE CONTAINS THE SYSTEM DIVISION AND THE MAIN ROUTINES    */
      6   /*  FOR THE MULTIFUNCTION KEYBOARD.                                   */
      7   /*                                                                    */
      8   /*  TASKS:          STARTER                                           */
      9   /*                  TIMEH                                             */
     10   /*                  FLASHSYMBOL                                       */
     11   /*                  KEYSCAN                                           */
     12   /*                                                                    */
     13   /*  PROCEDURES:     FIXKEY                                            */
     14   /*                  CLEARFORMAT                                       */
     15   /*                                                                    */
     16   /**********************************************************************/
     17
     18
     19   SYSTEM;
     20
     21   KEYBOARD:   -> STATICISER(2)*22;        /*THE KEYBOARD IS CONNECTED TO */
     22                                           /*CHANNEL 22 OF THE            */
     23                                           /*SECOND STATICISER.           */
     24   FUELSENSOR: -> STATICISER(2)*23;        /*CHANNEL 23.                  */
     25
     26   DISPLAY:    <- SPSERIALOUTPUT;          /*THE DISPLAY IS CONNECTED TO  */
     27                                           /*THE SPECIAL SERIAL OUTPUT.   */
     28
     29
     30   PROBLEM;
     31
     32   SPECIFY (AZ09LEG,ACTION) FIXED GLOBAL;
     33   SPECIFY FSYMBOL CHAR(1) GLOBAL;
     34   SPECIFY (TOGGLEORENTER,MFKEY,ALPHNUM,SYNC) SEMA GLOBAL;
     35   SPECIFY (DISPTIME,DISPFLSYMB) ENTRY (CHAR) GLOBAL,
     36           DISPLEGEND ENTRY (FIXED) GLOBAL;
     37   SPECIFY  NOW ENTRY RETURNS (CHAR) GLOBAL;
     38   SPECIFY (NUMERIC,LETTERAJ,LETTERKT,LETTERUZ,NAV,PLAN,CHECK,
     39            ROUTE,DEST,RB,FUEL,DISPPOS,
     40            NAVIGATION,SYSTEMTEST) TASK GLOBAL;
     41
     42   /*THE I/O DEVICE TYPES MUST BE SPECIFIED IN THE PROBLEM DIVISION     */
     43
     44   DECLARE KEYBOARD DEVICE SOURCE BIT(32);/* THE KEYBOARD SUPPLIES A    */
     45                                          /*BIT PATTERN (2 WORDS) .     */
     46   DECLARE FUELSENSOR DEVICE SOURCE FIXED;/* FUELSENSOR SUPPLIES AN     */
     47                                          /*INTEGER VALUE.              */
     48   DECLARE DISPLAY DEVICE SINK CHAR;      /* DISPLAY OUTPUT IS DONE BY A */
     49                                          /*SEQUENCE OF BYTES.          */
     50   DECLARE FORMLEG FIXED INIT(0);
     51            /*FORMLEG INDICATES THE FORMAT SELECTED                     */
     52            /*  = 0 . A FORMAT IS NOT YET SELECTED THEREFORE MULTI-      */
     53            /*        FUNCTION KEYS ARE INOPERATIVE                     */
     54            /*  = 8   FLAN  FORMAT                                      */
     55            /*  = 9   NAV   FORMAT                                      */
     56            /*  =10   CHECK FORMAT                                      */
     57
     58
     59
     60   STARTER: TASK GLOBAL;
```

# APPLICATIONS OF ASYNCHRONOUS MULTIPLEX DATA BUS IN THE TRIDENT TRAINING SITE

by

James Baker
General Electric

(This paper was not submitted)

MULTIPLEX DATA BUS CONFERENCE

10-12 OCTOBER 1978

DAYTON, OHIO

SESSION II

# SYSTEMS INTEGRATION USING MIL-STD-1553

MODERATOR: Colonel Eric B. Nelson
ESD/DCB
Hanscom AFB, MA

PRECEDING PAGE BLANK

# H-X
# NIGHT/ADVERSE - WEATHER AVIONICS STUDY

ASD/SDX

MRS. CHERYL L. ZELASCO



PRECEDING PAGE BLANK

# BRIEFING OBJECTIVE

- PRESENT MAJOR RESULTS OF NIGHT/ADVERSE WEATHER COMBAT RESCUE AVIONICS STUDY
  - PAVE LOW III CAPABILITY
  - H-60 CLASS HELICOPTER

# BACKGROUND



HH-53H Pave Low III



UH-60

# UH-60A AND HH-53H CHARACTERISTICS

| | UH-60A | HH-53H |
|---|---|---|
| WEIGHT (MAX) | 10,900 (20,250) | 23,500 (42,000) |
| CRUISE SPEED | 147 KTAS | 150 KTAS |
| DASH SPEED | 172 KTAS | 170 KTAS |
| RANGE (MAX GROSS WT.*) | 255 NM | 563 NM |
| CABIN CAPACITY | 11 TROOPS, 4 LITTERS | 37 TROOPS, 24 LITTERS |
| CABIN VOLUME | 396 CU. FT. | 1463 CU. FT. |
| AVIONICS VOLUME | 60 CU. FT. | 210 CU. FT. |

*PERFORMANCE COMPARISON @4000´ 95° DAY

# THE REQUIREMENTS

# VALIDATED REQUIREMENTS

## MAC GOR 4-77

REPLACE HH-3 SAR HELICOPTER

COMBAT AIRCREW RECOVERY IN HOSTILE TERRITORY

NIGHT RESCUE IN ADVERSE WEATHER

SINGLE SHIP, COVERT OPERATION

PAVE LOW III PERFORMANCE

UH-60 CLASS AIR VEHICLE

SURVIVABILITY



# THE STUDY

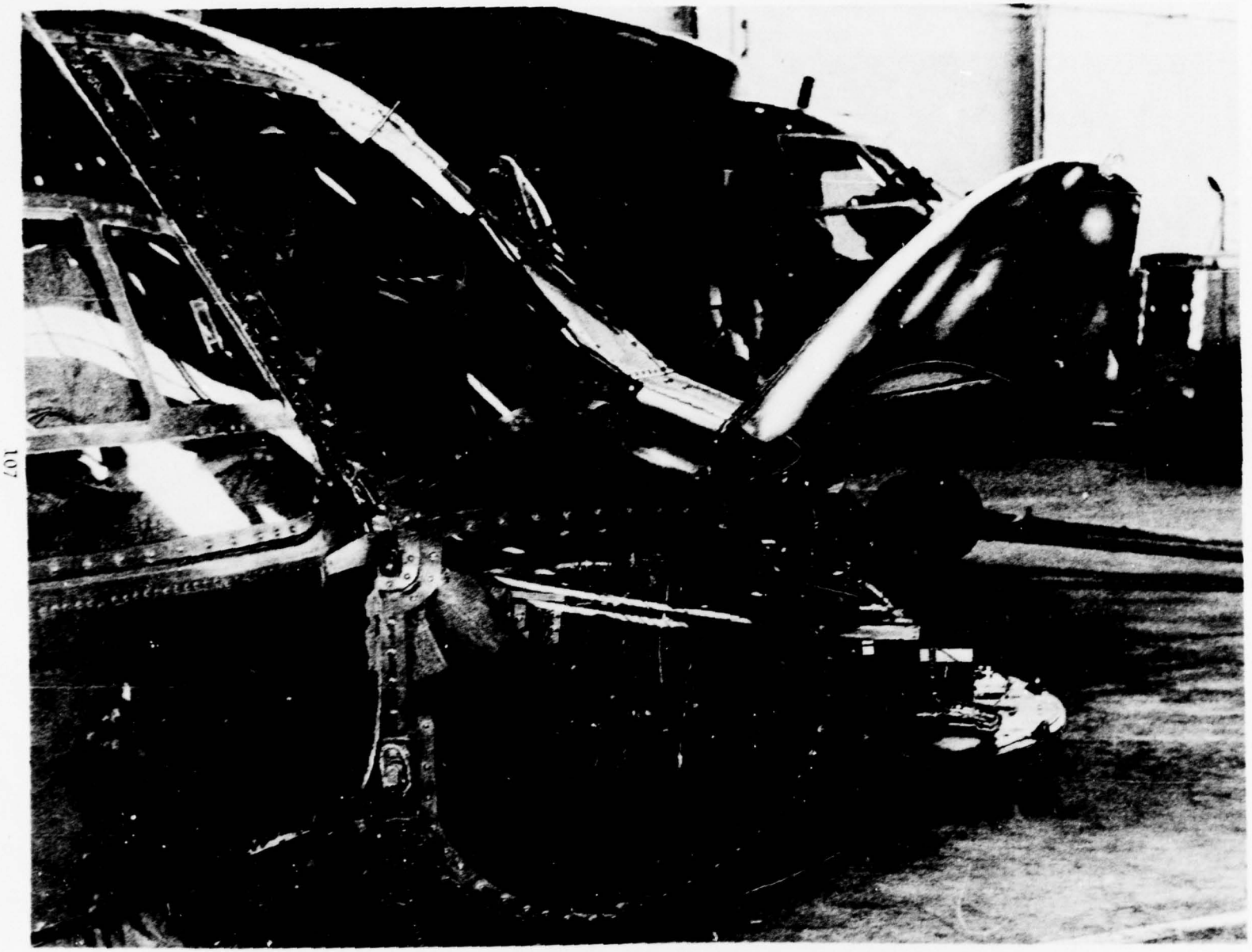
# THE ASD STUDY TEAM

## SDX

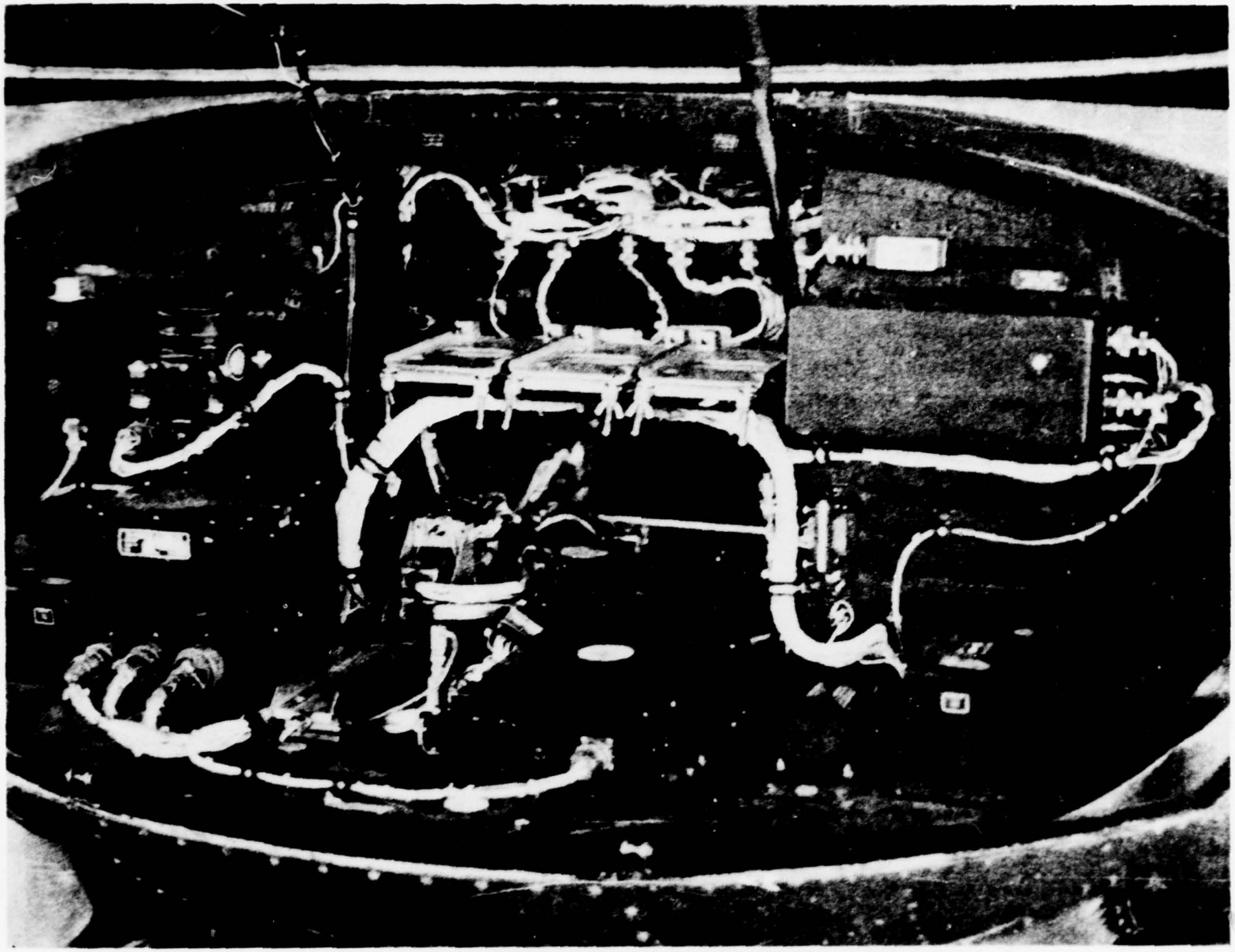
| | |
|---|---|
| PROJECT MANAGER | MAJOR A. LEATHAM |
| ENGINEERING OFFICER | CAPTAIN K. LEONE |
| TECHNICAL CONSULTANT | MR. J. GRAVES |

## ENA

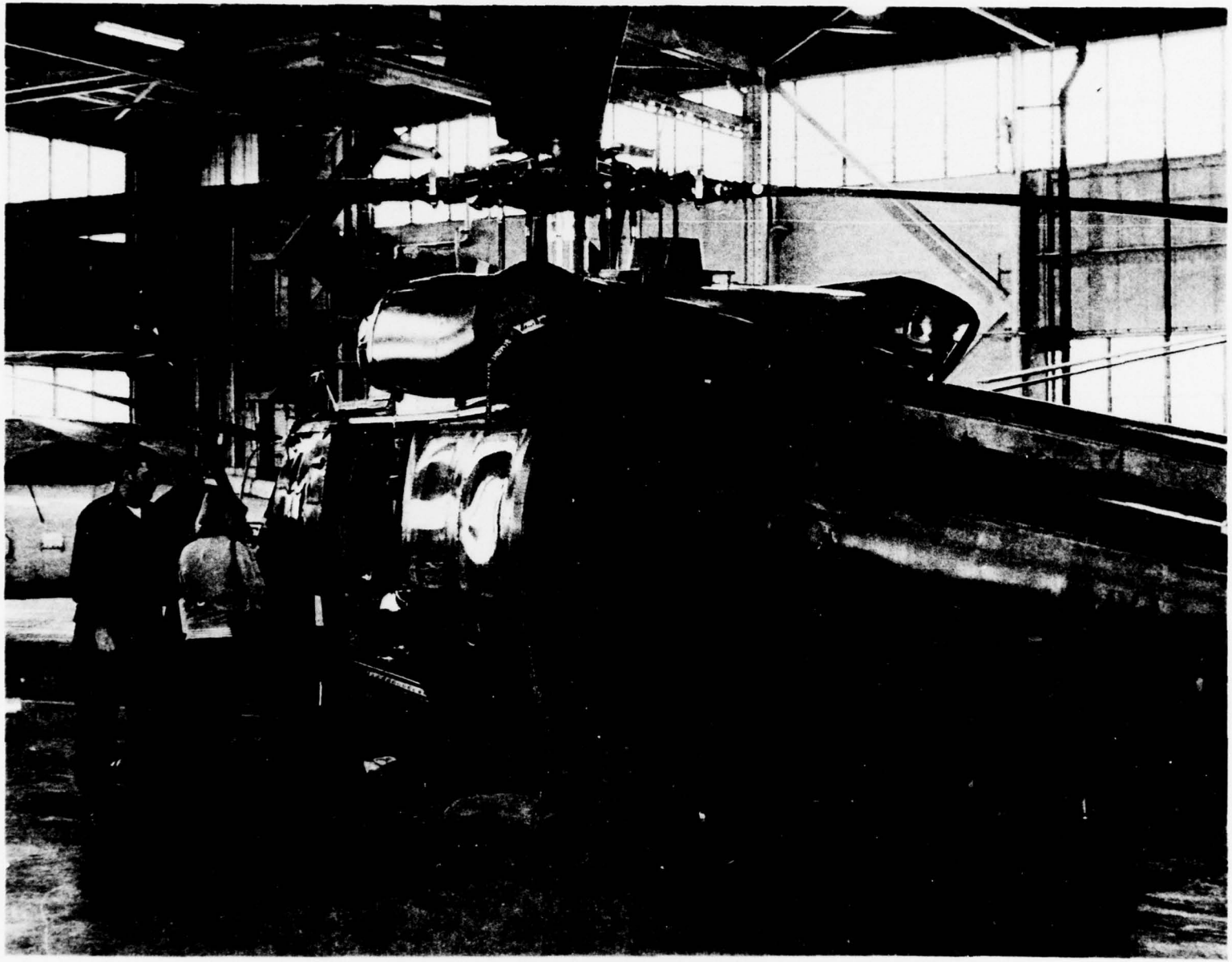
| | | | |
|---|---|---|---|
| STUDY CHAIR | MRS. C. ZELASCO | RADAR | MR. N. GOFF |
| ALT. STUDY CHAIR | MR. F. GLOVER | RADAR | DR. D. KRILE |
| INTEGRATION | MR. M. ENGLE | FLIR | MR. R. NICKOLS |
| DIGITAL HARDWARE | CAPTAIN R. PURVIS | LASERS | CAPTAIN S. SHINKLE |
| DIGITAL HARDWARE | LT. D. LYONS | PROTECTIVE SYSTEMS | MR. M. BARNHART |
| SOFTWARE | LT. R. SWARTZ | INSTRUMENTS | MR. C. SWEET |
| IFF/TACAN/ILS | MR. R. STANFORD | DISPLAYS | MR. J. EIDENMILLER |
| COMMUNICATIONS | MR. R. MINOR | EMIC | MR. K. KOO |
| COMMUNICATIONS | LT. A. KISMERIZAK | EMIC | MR. D. BASELEY |
| NAVIGATION | MR. C. McCAMEY | SYSTEM ANALYSIS | MR. J. PATTERSON |



UH-60 SIDE NOSE



UH-60 NOSE AVIONICS BAY



UH-60 REAR



UH-60 LEFT SIDE COCKPIT 110

# AVIONICS SCREENING FACTORS

PERFORMANCE

COST

RELIABILITY

PRODUCIBILITY

AIR VEHICLE COMPATIBILITY

MAINTAINABILITY

SURVIVABILITY

FLEXIBILITY/COMMONALITY

# TECHNOLOGY SURVEY RESULTS

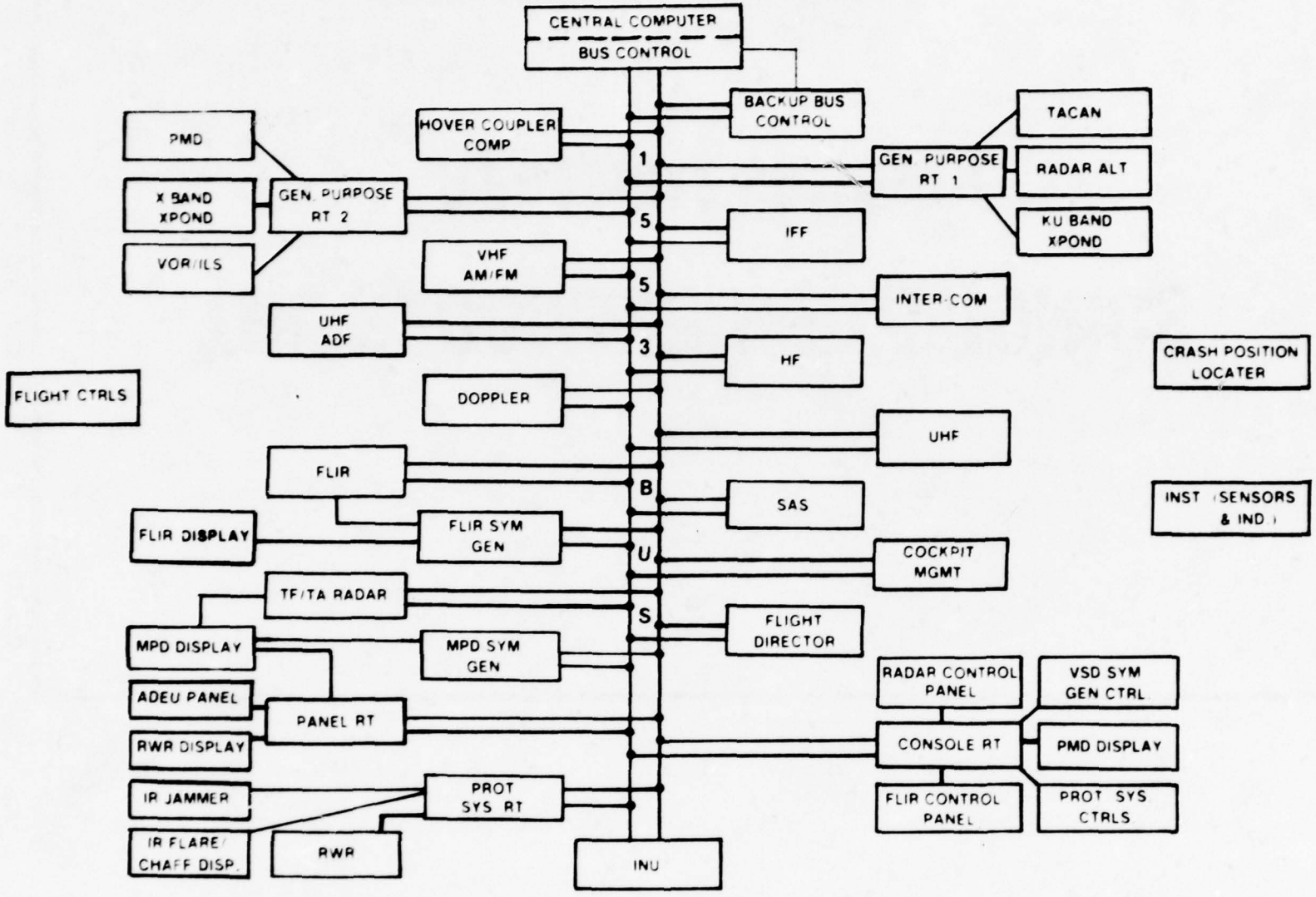
- NO TECHNOLOGY DEFICIENCIES TO PRODUCE RECOMMENDED H-X AVIONICS
- SOME EQUIPMENT NOT IN INVENTORY-INDUSTRY CAN PRODUCE
- POSSIBLE GROWTH POTENTIAL IN PRESENTLY IMMATURE TECHNOLOGY AREAS



# THE RECOMMENDED AVIONICS

## A POSSIBLE SOLUTION

# H-X AVIONICS FUNCTIONAL BLOCK DIAGRAM - RECOMMENDED APPROACH



# H-X ARCHITECTURE FEATURES

- MIL-STD-1553 DATA BUS
  - DUAL REDUNDANT
  - BUS LOADING = 9 PERCENT
- CENTRAL COMPUTER WITH BUS CONTROLLER
- STAND ALONE BACKUP BUS CONTROLLER
- DATA BUS/SUBSYSTEM INTERFACES
  - INTEGRAL TO SUBSYSTEM
  - REMOTE TERMINALS
- INTEGRATED CONTROLS AND DISPLAYS
- PARTITIONING OF AVIONICS BETWEEN NOSE & AFT BAYS
- MODIFICATION AND GROWTH FLEXIBILITY

# H-X AVIONICS FUNCTIONAL AREAS

- COMMUNICATIONS
- RADIO NAVIGATION AIDS
- PROTECTIVE SYSTEMS
- PRECISION NAVIGATION AND SURVIVOR LOCATION
- TERRAIN FOLLOWING AND NIGHT VISION SENSORS
- COMPUTER
- DISPLAYS
- INSTRUMENTS

# RECOMMENDED COMMUNICATIONS

| SUBSYSTEM | LOCATION IN H-60 | CONTROL FCTNS | DISPLAY FCTNS |
|---|---|---|---|
| UHF RADIO | AFT TRANS BAY | CMS | CMS/ICS/MPD |
| VHF AN/FM RADIO | AFT TRANS BAY | CMS | CMS/ICS/MPD |
| HF RADIO | AFT TRANS BAY | CMS | CMS/ICS/MPD |
| INTERCOM (ICS) | CENTER CONSOLE & CABIN | ICS | ICS |
| SECURE VOICE ENCRYPTION | AFT TRANS BAY | CMS/MECH CODE ENTRY | CMS/ICS |
| IFF (WITH SECURE MODE) | AFT TRANS BAY | CMS/MECH CODE ENTRY | CMS/ICS/MPD |
| X BEACON | AFT TRANS BAY | CMS | CMS |
| Ku BEACON | AFT TRANS BAY | CMS | CMS |
| CRASH POSITION INDICATOR | AFT EXT MOUNT | DEDICATED SWITCH | N/A |
| LOUD HAILER COMPATIBILITY | N/A | N/A | N/A |

MPD - MULTI-PURPOSE DISPLAY

ICS - INTERCOM

CMS - COCKPIT MANAGEMENT SYSTEM

# RECOMMENDED RADIO NAVIGATION AIDS

| SUBSYSTEM | LOCATION IN H-60 | CONTROL FCTNS | DISPLAY FCTNS |
|---|---|---|---|
| UHF/ADF | AFT TRANS BAY | CMS | CMS/INST/MPD/ICS |
| VOR/ILS | AFT TRANS BAY | CMS | CMS/INST/MPD/ICS |
| TACAN | AFT TRANS BAY | CMS | CMS/INST/MPD/ICS |
| GPS (GROWTH) | AFT TRANS BAY | CMS | CMS/MPD |

CMS - COCKPIT MANAGEMENT SYSTEM

ICS - INTERCOM

INST - INSTRUMENTS

MPD - MULTIPURPOSE DISPLAY

# RECOMMENDED PROTECTIVE SYSTEMS

| SUSBYSTEM | LOCATION IN H-60 | CONTROL FCTNS | DISPLAY FCTNS |
|---|---|---|---|
| RADAR WARNING RECEIVER (ECM) | AFT TRANS BAY | ECM CTRL PANELS | RWR DISPLAY |
| CHAFF AND FLARE DISPENSER (ECM, IRCM) | AFT EXT MOUNT | ECM CTRL PANELS | N/A |
| IR JAMMER (IRCM) | AFT EXT MOUNT | ECM CTRL PANELS | N/A |
| IR ENGINE EXHAUST SUPPRESSOR (IRCM) | ENGINES | N/A | N/A |

RWR - RADAR WARNING RECEIVER

ECM - ELECTORNIC COUNTERMEASURES

IRCM - INFRARED COUNTERMEASURES

* DEVELOPMENT RESPONSIBILITY PER JTCG-AS

ARMY - LIGHT LIFT HELICOPTERS

NAVY - HEAVY LIFT HELICOPTERS

# RECOMMENDED PRECISION NAVIGATION AND SURVIVOR LOCATION

| SUBSYSTEM | LOCATION IN H-60 | CONTROL FCTNS | DISPLAY FCTNS |
|---|---|---|---|
| INS (INCL RNAV) | AFT TRANS BAY | CMS | CMS/MPD/FLIR VSD |
| RNAV AUTO DATA ENTRY UNIT (ADEV) | CENTER CONSOLE | CMS & ADEU | CMS |
| DOPPLER RADAR | AFT EQUIP BAY | CMS | CMS/MPD/FLIR VSD |
| RADAR ALTIMETER | AFT EQUIP BAY | CMS | CMS/MPD/FLIR VSD |
| SURVIVOR LOCATION | AFT EQUIP BAY | CMS | CMS/MPD |

CMS - COCKPIT MANAGEMENT SYSTEM

FLIR VSD - FLIR, VERTICAL SITUATION DISPLAY

MPD - MULTIPURPOSE DISPLAY

# RECOMMENDED
# TF/TA AND NIGHT VISION SENSORS

| SUBSYSTEM | LOCATION IN H-60 | CONTROL FCTNS | DISPLAY FCTNS |
|---|---|---|---|
| TF/TA MULTIMODE RADAR | NOSE & AFT TRANS BAY | RADAR CTRL PANEL | MPD/FLIR/VSD /INST |
| FLIR | NOSE & AFT TRANS BAY | FLIR CTRL PANEL | FLIR VSD |

INST - INSTRUMENTS

MPD - MULTI-PURPSOSE DISPLAY

FLIR VSD - FLIR, VERTICAL SITUATION DISPLAY

# RECOMMENDED COMPUTER

| SUBSYSTEM | LOCATION IN H-60 | CONTROL FCTNS | DISPLAY FCTNS |
|---|---|---|---|
| CENTRAL COMPUTER | AFT TRANS BAY | CMS | CMS |
| BACK-UP BUS CONTROLLER | AFT TRANS BAY | CMS | CMS |
| CONSOLE REMOTE TERMINAL | UNDER CENTER CONSOLE | N/A | N/A |
| PROTECTIVE SYSTEMS REMOTE TERMINAL | AFT TRANS BAY | N/A | N/A |
| GENERAL PURPOSE REMOTE TERMINALS (2) | AFT TRANS BAY | N/A | N/A |
| PANEL REMOTE TERMINAL | NOSE EQUIP SHELF | N/A | N/A |
| FLIGHT DIRECTOR (FD) | NOSE EQUIP SHELF | FD PANELS | CMS/MPD /INST |

CMS - COCKPIT MANAGEMENT SYSTEM
MPD - MULTIPURPOSE DISPLAY
INST - INSTRUMENTS

# RECOMMENDED DISPLAYS

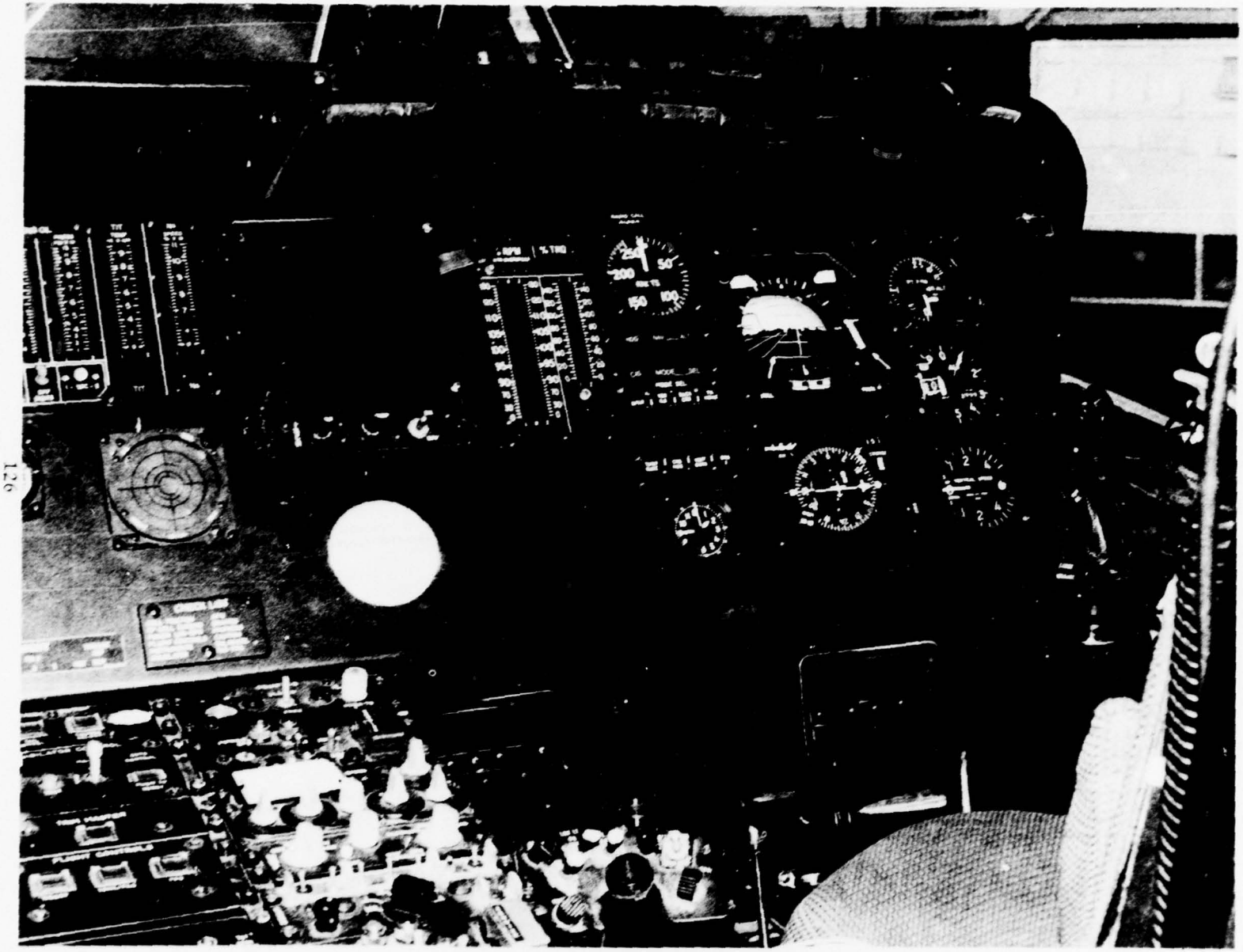
| SUBSYSTEM | LOCATION IN H-60 | CONTROL FCTNS |
|---|---|---|
| FLIR/VERT SIT DISPLAY (2) (VSD) | INST PANEL | FLIR CTRL PANEL |
| VSD SYMBOL GENERATOR | AFT TRANS BAY | FLIR CTRL PANEL |
| MULTIPURPOSE DISPLAY (2) (MPD) | INST PANEL | CMS/MPD |
| MPD SYMBOL GENERATOR | AFT TRANS BAY | MPD |
| MPD DIGITAL SCAN GENERATOR | AFT TRANS BAY | MPD |
| PROJECTED MAP DISPLAY (PMD) | CENTER CONSOLE | PMD |
| RADAR WARNING RECEIVER DISPLAY | INST PANEL | ECM CTRL PANEL |
| COCKPIT MANAGEMENT SYSTEM (2) (CMS) | CENTER CONSOLE | CMS |

# RECOMMENDED INSTRUMENT SENSORS

| SUBSYSTEM | LOCATION IN H-60 |
|---|---|
| LOW AIRSPEED TRANSDUCER | ROTOR TOP MOUNT |
| OUTSIDE AIR TEMPERATURE | EXTERNAL MOUNT |
| VERTICAL GYROS (2) | UH-60 STANDARD |
| DIRECTIONAL GYRO | UH-60 STANDARD |
| SIGNAL DATA CONV (PROPULSION) | UH-60 STANDARD |
| AIRSPEED TRANSDUCER (2) | UH-60 STANDARD |
| PITOT-STATIC PROBE (2) | UH-60 STANDARD |
| FUEL QTY SENSOR & TRANSMITTERS | UH-60 STANDARD |
| TRANSMISSION OIL PRESSURE & TEMP (2) | UH-60 STANDARD |
| ENGINE OIL PRESSURE & TEMP (2) | UH-60 STANDARD |
| TURBINE INLET TEMP (2) | UH-60 STANDARD |
| POWER TURBINE TACHOMETER (2) | UH-60 STANDARD |
| ROTOR TACHOMETER | UH-60 STANDARD |
| MAGNETIC AZIMUTH DETECTOR | UH-60 STANDARD |

# RECOMMENDED PANEL INSTRUMENTS

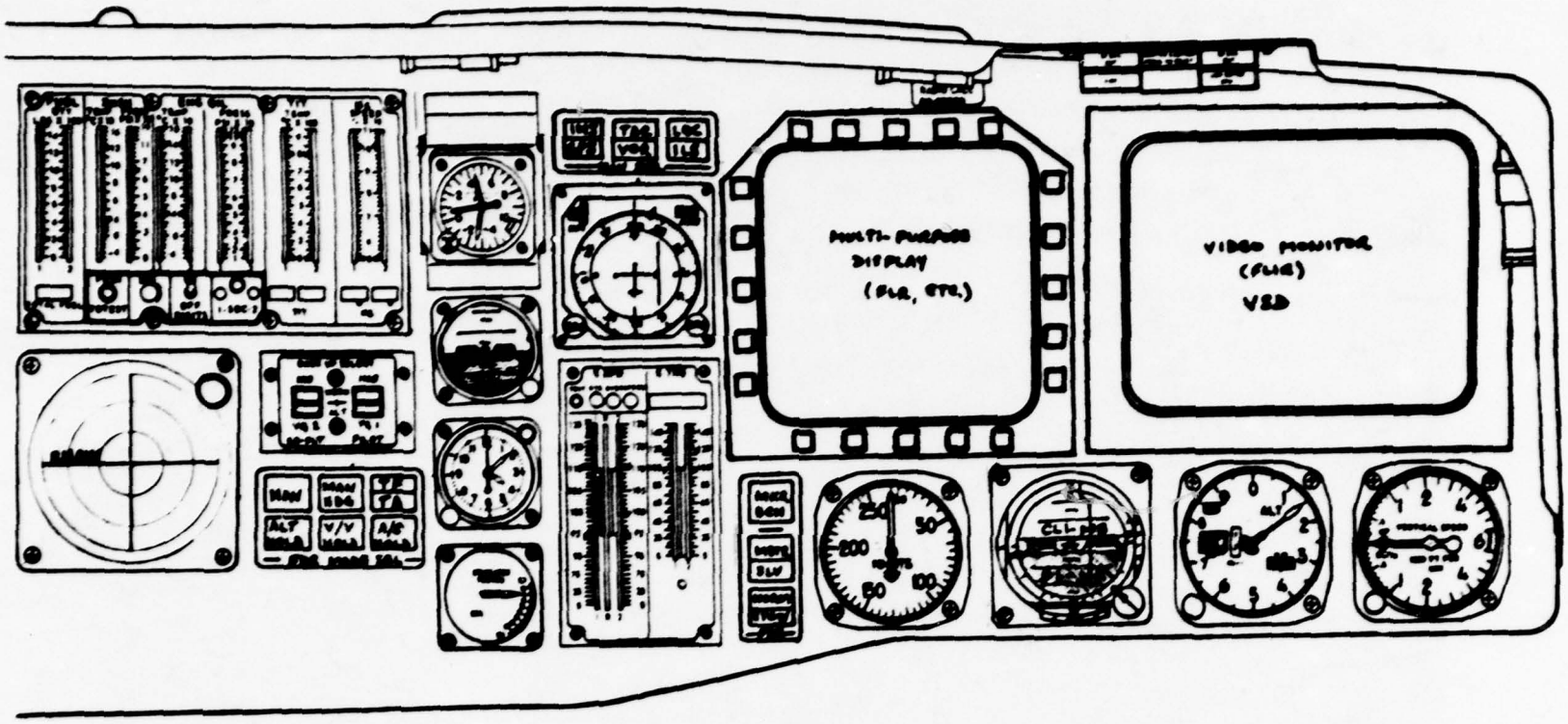
| SUBSYSTEM | LOCATION IN H-60 |
|---|---|
| 3'' ADI (2) | INST PANEL |
| 3'' HSE (2) | INST PANEL |
| 2'' ATTITUDE INDICATOR (2) | INST PANEL |
| FLIGHT DIRECTOR CONTROL PANELS | INST PANEL |
| BARO ALTIMETER (2) | INST PANEL |
| AIRSPEED INDICATOR (2) | INST PANEL |
| VERTICAL SPEED INDICATOR (2) | INST PANEL |
| RPM & TORQUE IND | INST PANEL |
| ACCELEROMETER (G METER) | INST PANEL |
| ATTITUDE INDICATOR (2'') | INST PANEL |
| CLOCK, 8-DAY | INST PANEL |
| PROPULSION INSTR/DISPLAY | INST PANEL |
| MASTER CAUTION PANEL (2) | INST PANEL |
| WARNING/CAUTION PANEL | INST PANEL |
| STABILATOR POSITION INDICATOR | INST PANEL |
| MAGNETIC OMPASS STDBY | INST PANEL |



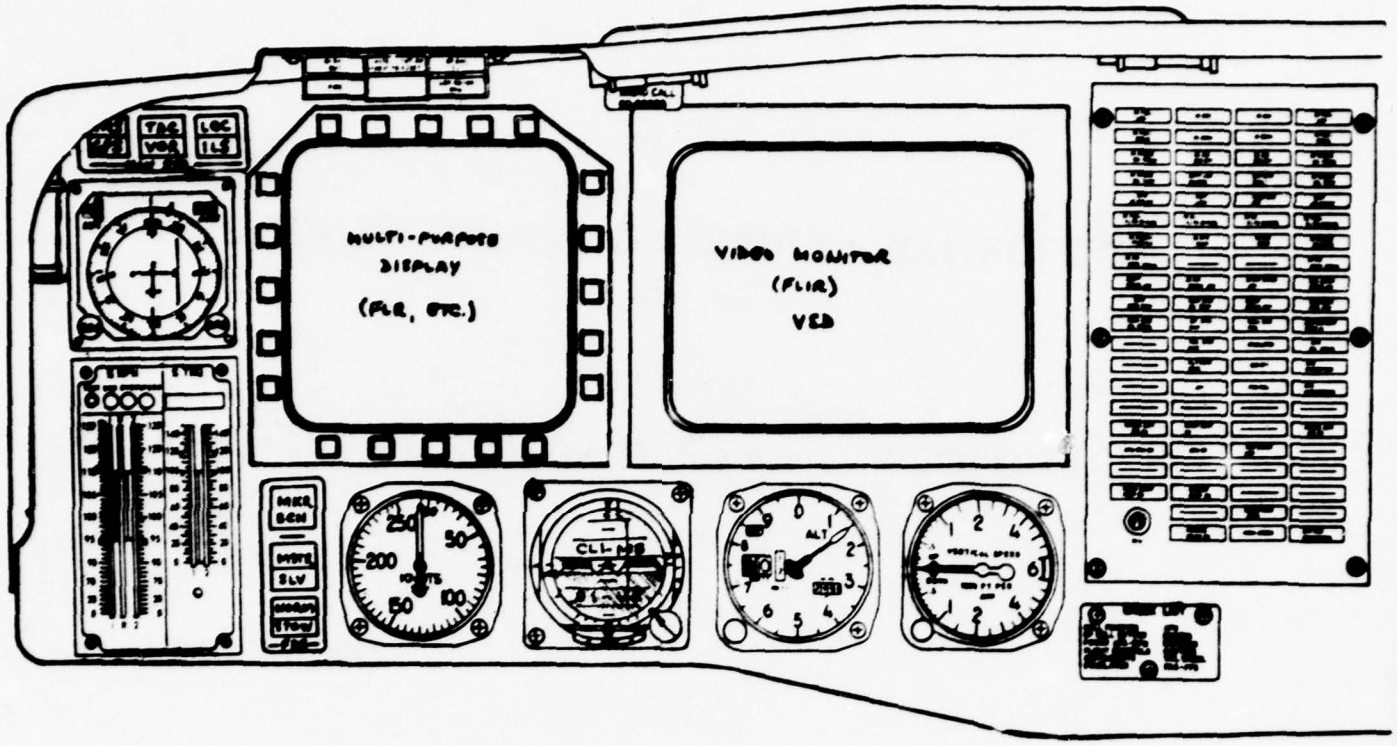
UH-60 RIGHT INSTRUMENT PANEL



UH-60 LEFT INSTRUMENT PANEL




RECOMMENDED RIGHT INSTRUMENT PANEL




RECOMMENDED LEFT INSTRUMENT PANEL

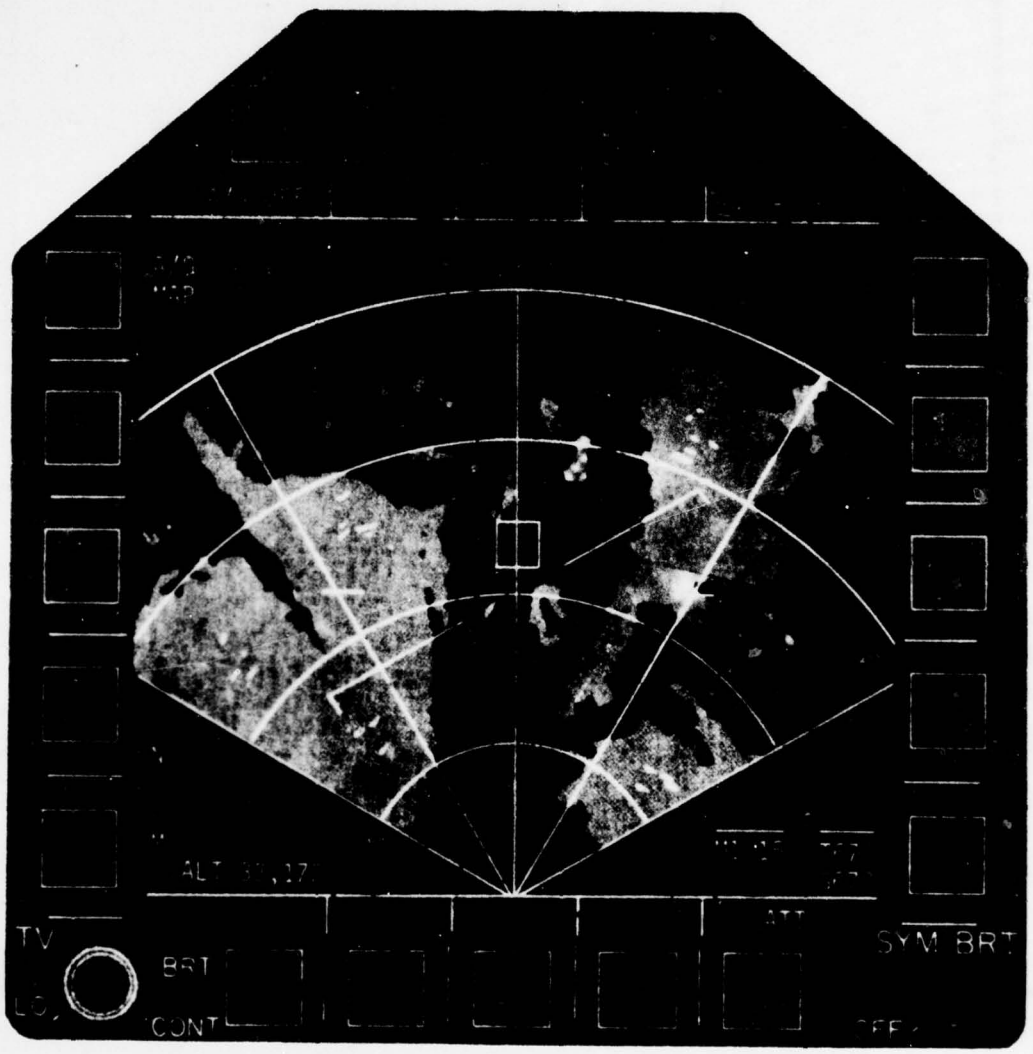
# RECOMMENDED FLIR/VERTICAL SITUATION DISPLAY FUNCTIONS

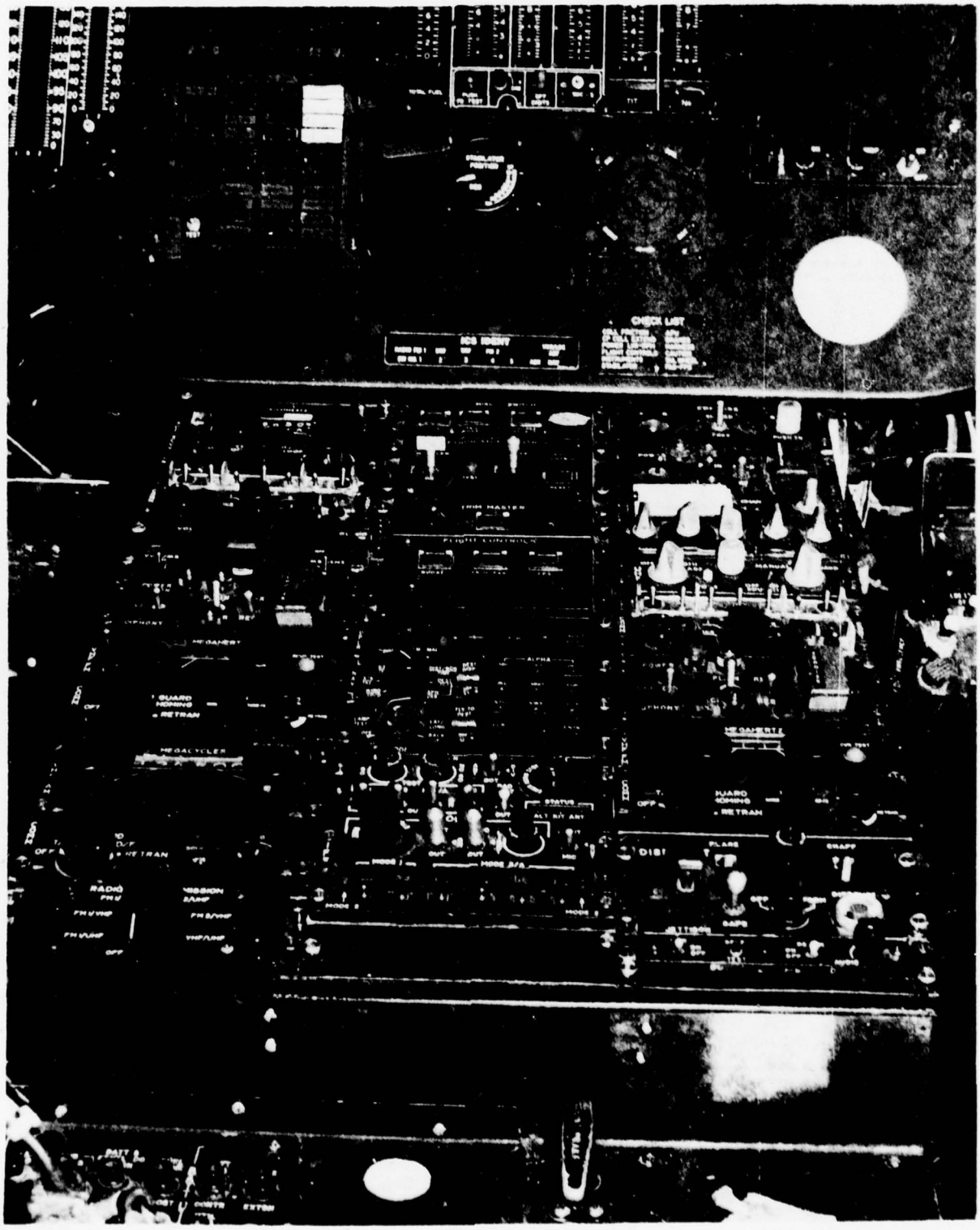
- FLIR VIDEO
- VERTICAL SITUATION DISPLAY SYMBOLOGY
- HORIZONTAL SITUATION DISPLAY SYMBOLOGY

# RECOMMENDED MULTIPURPOSE DISPLAY FUNCTIONS

- RADAR DISPLAY
  - PPI
  - $E^2$
  - FLIGHT DIRECTOR
- NAVIGATION DISPLAYS
  - FLIGHT PLAN ROUTE
  - EHSI
  - DIGITAL READOUT
- SURVIVAL AVIONICS SYSTEM
- BACKUP DISPLAY FOR FLIR/VSD
- ENGINE & HOVER PERFORMANCE
- FUTURE SYSTEMS
  - JTIDS
  - ETC.

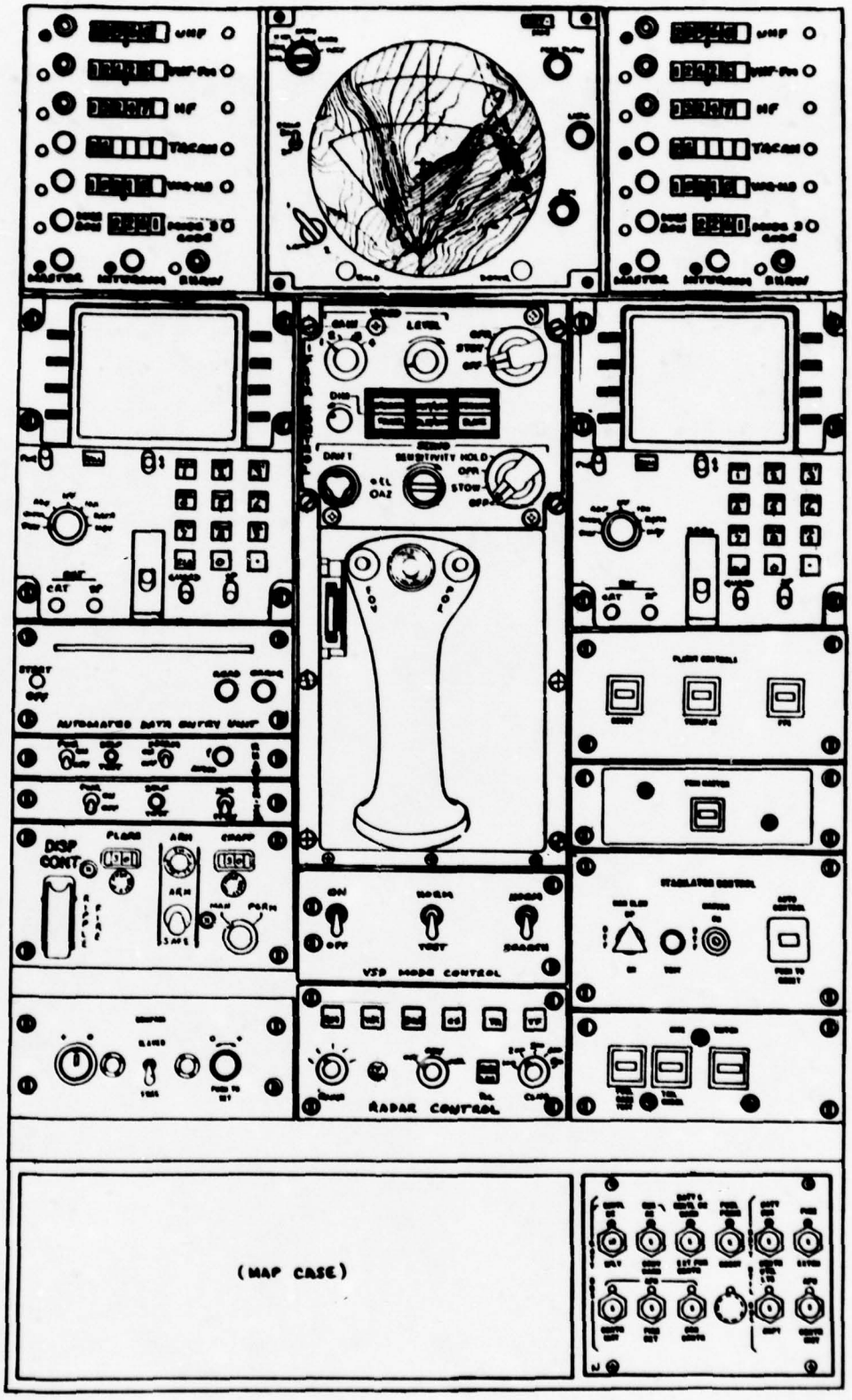
## GROUND MAP






UH-60 CENTER CONSOLE 133

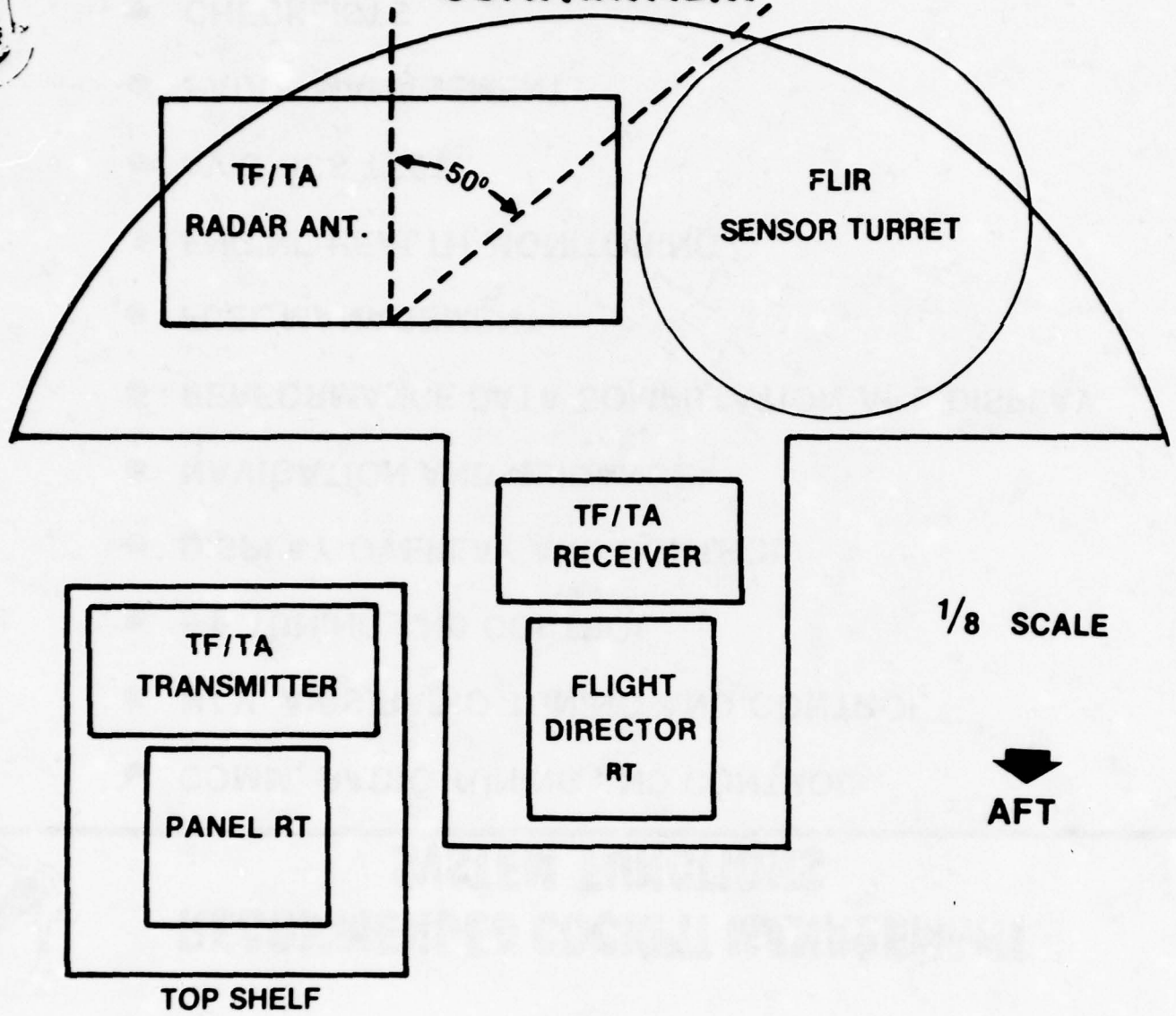
RECOMMENDED CENTER CONSOLE



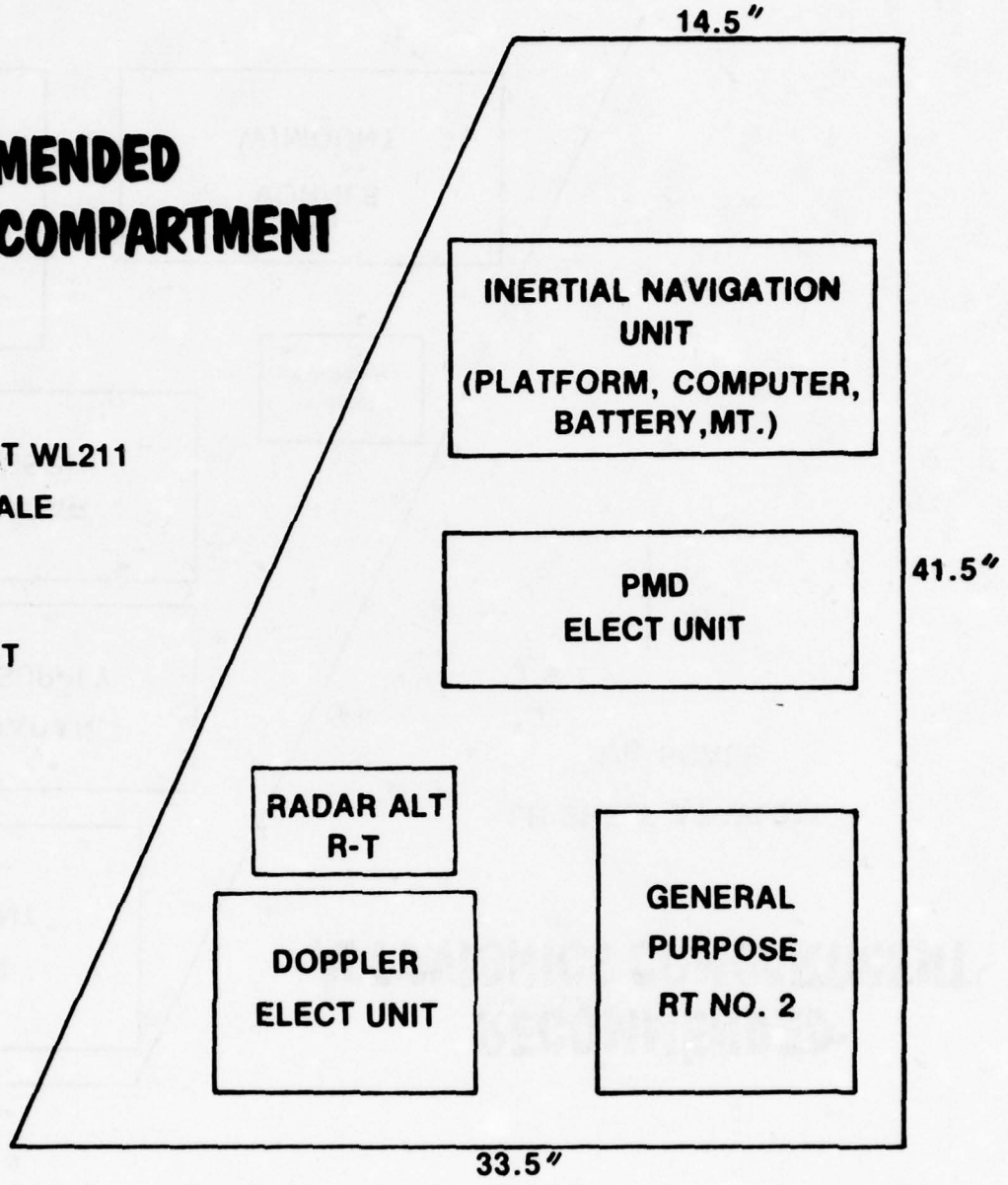
# RECOMMENDED COCKPIT MANAGEMENT SYSTEM FUNCTIONS

- COMM. RADIO TUNING AND CONTROL
- NAV. AIDS RADIO TUNING AND CONTROL
- IFF TUNING AND CONTROL
- DISPLAY OVERLAY AND CONTROL
- NAVIGATION AND GUIDANCE
- PERFORMANCE DATA COMPUTATION AND DISPLAY
- FUEL MANAGEMENT
- ENGINE HEALTH MONITORING
- AVIONICS TEST
- FAULT MANAGEMENT
- CHECKLISTS

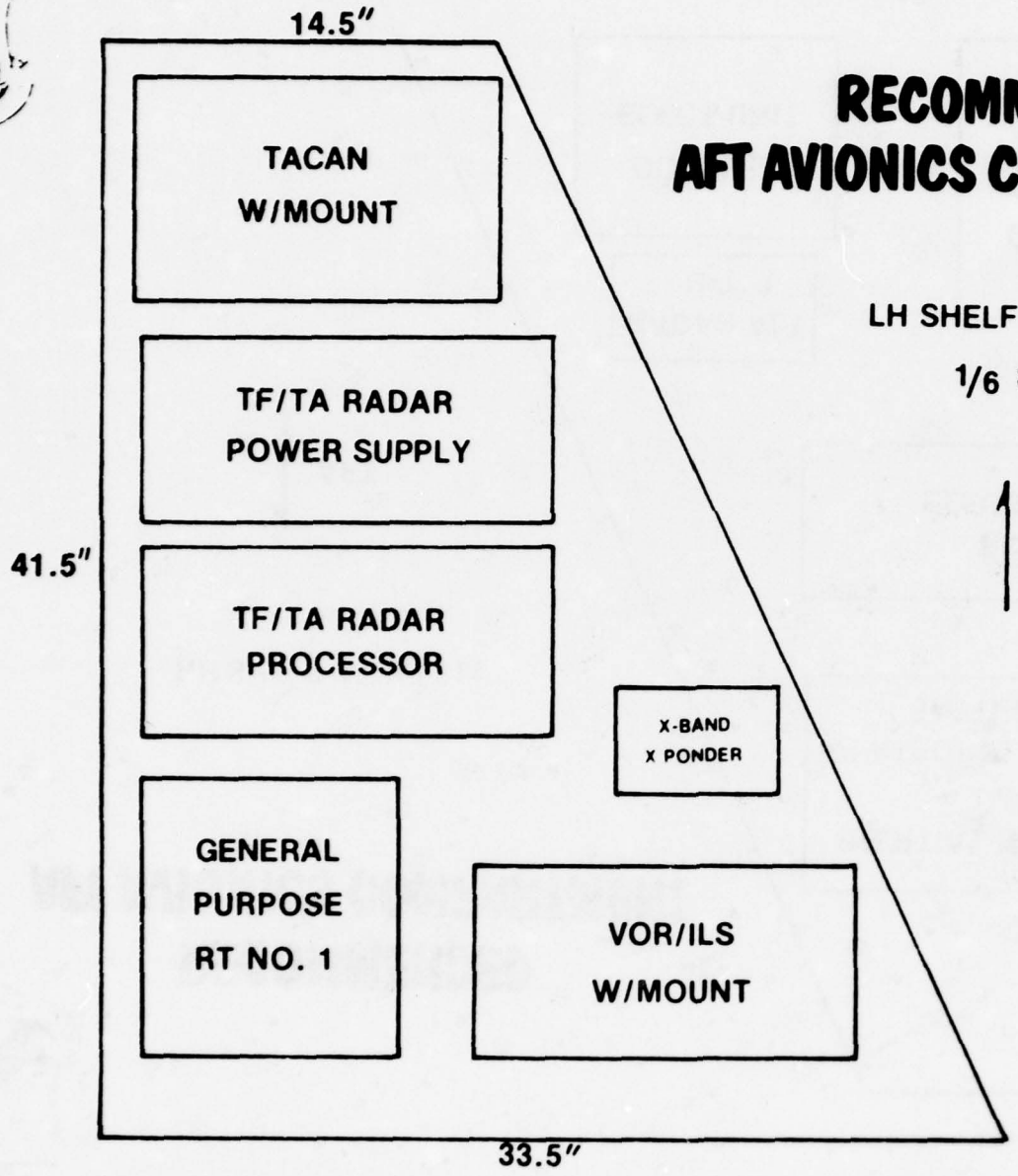
# RECOMMENDED NOSE FAIRING AND ELECTRONICS COMPARTMENT



# RECOMMENDED
# AFT AVIONICS COMPARTMENT



# RECOMMENDED AFT AVIONICS COMPARTMENT

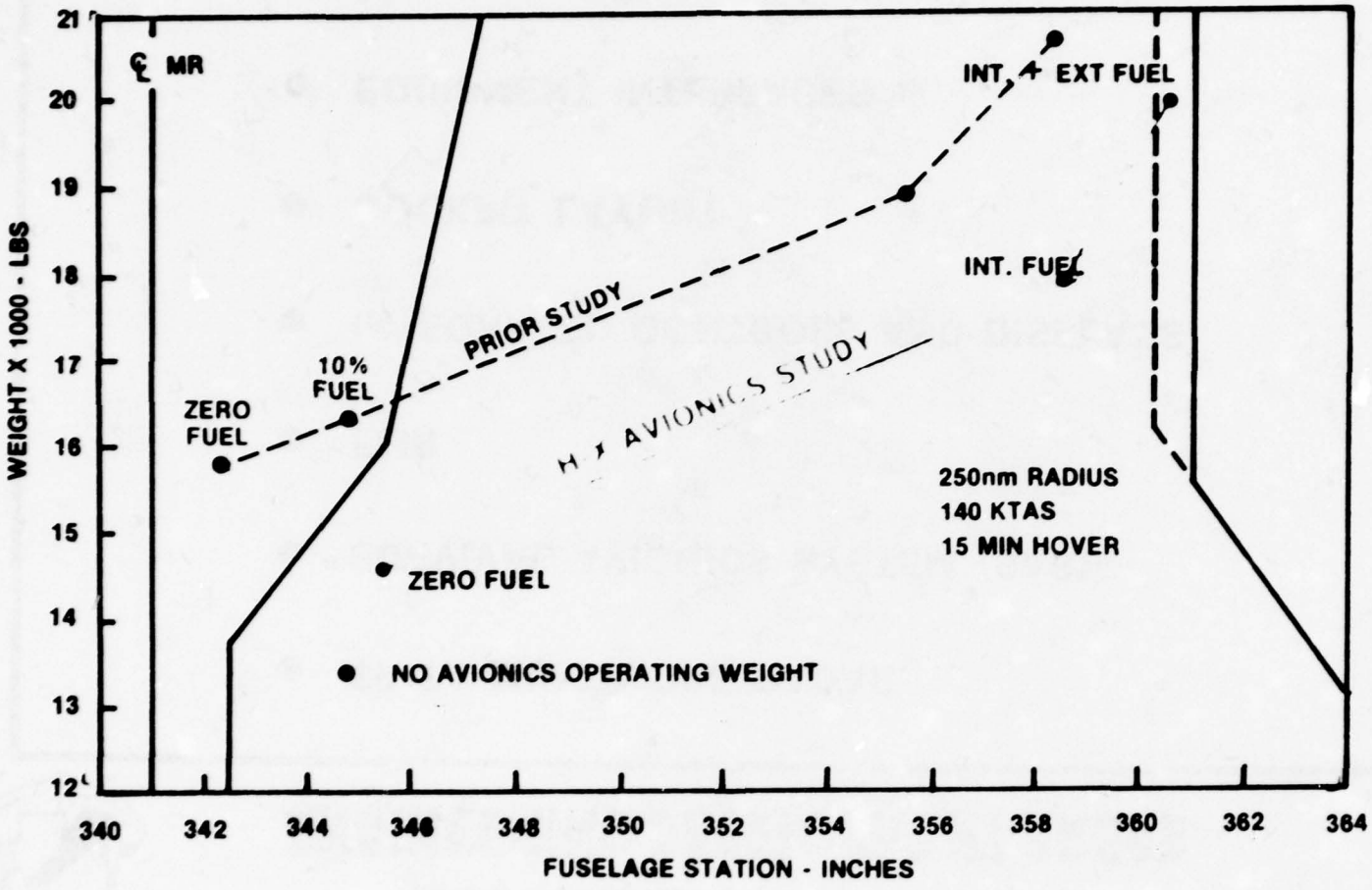



LH SHELF AT WL211

1/6 SCALE

AFT

# H-X C.G. WITHIN LIMITS






H-X AVIONICS STUDY

# RECOMMENDED AVIONICS ENGINEERING DEVELOPMENT AREAS

- TF/TA MULTIMODE RADAR
- SURVIVAL AVIONICS SYSTEM (SAS)
- FLIR
- INTEGRATED CONTROLS AND DISPLAYS
- COCKPIT LAYOUT
- EQUIPMENT INTERFACES
- COMPUTER SOFTWARE

# RECOMMENDED AVIONICS FEATURES

| | GOAL | RECOMMEDED CONFIGURATION |
|---|---|---|
| WEIGHT | 1500 LBS INCL 25% FOR INSTALLATION | 1200 LBS + INSTALLATION TBD |
| CG | VARIABLE ABOUT FUSELAGE STATION 344.2 | ACHIEVED GOAL |
| MTBF | 150 HOURS | EXPECT TO ACHIEVE GOAL |
| COST | MINIMIZE | EQUIPMENT + INTEGRATION TBD + NON-RECURRING TBD |
| PERFORMANCE | PAVE LOW III | ACHIEVED GOAL |
| SIZE | NOSE, PANEL, CONSOLE + AFT TRANSITION COMPARTMENT FIT | ACHIEVED GOAL |
| DEVELOPMENT | MINIMIZE | MULTIMODE TF/TA RADAR |

# HOW MIL-STD-1553 HELPS H-X

- PERMITS INTEGRATION OF CONTROLS & DISPLAYS IN SMALL COCKPIT
- ALLOWS NOSE BAY TO BE LIMITED PRIMARILY TO SENSORS
- HELPS SOLVE WEIGHT AND BALANCE PROBLEM
- HELPS MINIMIZE SIZE & WEIGHT OF WIRE BUNDLES
- AIDS MODIFICATION AND GROWTH
- REDUCES EMC EFFORT



# STUDY SUMMARY

- NIGHT ADVERSE WEATHER SAR CAPABILITY IN UH-60 IS PRACTICAL
- TF/TA RADAR LOW RISK DEVELOPMENT IS IMPERATIVE
- FLIR TURRETT LOW RISK MODIFICATION IS REQUIRED
- INTEGRATED CONTROLS AND DISPLAYS
- 1553A DATA BUS REQUIRED
- PERFORMANCE VS. COST OPTIONS ARE AVAILABLE
- POSSIBLE GROWTH POTENTIAL IN PRESENTLY IMMATURE TECHNOLOGY AREAS

# MULTIPLEX SYSTEM FOR THE HUGHES ADVANCED ATTACK HELICOPTER - YAH-64

Frank Booth
Supervisor, Multiplex Group
Hughes Helicopters
Culver City, California

David Espen
Lead Engineer, Advanced Projects
Sperry Flight Systems
Phoenix, Arizona

## Abstract

The multiplex system, as configured in the YAH-64 helicopter, performs the function of command and control of avionic and mission equipment functions through acquisition, conditioning, processing, and channeling of weapon system data for over 1,300 signals.

The YAH-64 multiplex system approach brings the advantages of functional and configurational flexibility to the aircraft while simultaneously reducing production costs and increasing reliability. In addition, integrated within the YAH-64 multiplex system is the Fault Detection/Location Subsystem (FD/LS), which provides the flight crew and maintenance personnel with aircraft subsystem operational status.

This paper discusses the design features, implementation, and operation of the YAH-64 multiplex system. Overall system operational features, redundancy, and design features will be discussed.

## Introduction

The YAH-64 Advanced Attack Helicopter, shown in Figure 1, is being developed for the U.S. Army. Using a multiplexing system to transfer data and provide controls for avionic and mission equipments, the YAH-64 multiplex system makes extensive use of monolithic and hybrid components, including integrated automatic built-in test circuitry and system redundancy. Conforming to MIL-STD-1553A, which is the military standard for

Aircraft Internal Time Division Multiplex Data Bus Systems, the YAH-64 multiplex system makes this the first helicopter to incorporate an integrated avionics/weapons system utilizing a multiplex data bus system for subsystem interface and data handling.

Some of the major functions performed through the YAH-64 multiplex system include:

- Mode control and display data handling for the rocket, missile, target acquisition, pilot night vision, integrated helmet sights displays, and electronic attitude indicator subsystems.
- Video selection
- Provides flexible interface for avionic and mission equipment
- Fault detection and location

The present YAH-64 multiplex system contains a 50 percent growth capability in the areas of memory, subsystem interfaces and bus utilization time. Aircraft weapon control and display operation can be effectively modified through reprogramming to accommodate system reconfigurations for future requirements with minimum cost and schedule impacts.

System Description

The YAH-64 multiplex system is a distributed time division multiplex system consisting of 13 units, interfacing directly to the redundant data busses. These 13 units process more than 1,300 signal inputs and outputs, scattered throughout the vehicle. Of the 13 bus interfacing units, nine are Remote Terminal Units specifically designed to adapt off-bus subsystems to the multiplex data bus. Where possible, interfaces to Remote Terminal Units have been standardized as discrete (bi-level) ac and dc analog, synchro, and serial digital data. Functionally, the YAH-64 multiplex system replaces much of the signal/control wire and relay logic required in conventional aircraft configurations. The system can be expanded to include 32 units in order to meet future requirements.

Figure 2 is the block diagram of the present YAH-64 multiplex system. The system consists of:

- Dual data busses
- Two bus controllers (primary residing in the fire control computer; backup residing in the copilot compartment remote terminal unit).
- Symbol generator

- Missile system remote electronics unit
- Electronic attitude director indicator remote electronics unit
- Four general-purpose remote terminal units located in the pilot's compartment, right forward avionics bay, left forward avionics bay, and aft avionics bay.
- Four pylon remote terminal units (one located in each pylon).
- Twenty-six data link termination units

Figure 3 illustrates the YAH-64 multiplex system box distribution throughout the aircraft. The primary data bus is routed on the left side of the aircraft, while the secondary data bus is routed on the right side. This isolation between the busses increases survivability. Both bus controllers (FCC and BBC) are widely separated for the same reason. Critical signals can be routed into separate remote terminal units by providing separate signal paths, precluding the loss of that function with a remote terminal malfunction.

Data Bus

The YAH-64 multiplex system data bus operates asychronously in a command/response mode, with transmissions occurring in a half-duplex manner. Each of the multiplex system data busses consist of a low loss twisted shielded, 24 guage, teflon insulated wire pair, terminated at each end in its characteristic impedance (71 ohms nominal). All connections to the data bus system utilize a small (.1 lb) Data Link Termination Unit (DLT) as illustrated in Figure 4. The Data Link Termination Units provide the bus with short-circuit isolation, impedance matching, and line termination. One DLT per bus is used for each terminal, thereby requiring two DLTs per terminal for the dual redundant system.

Information flow on the data bus is comprised of messages and words per MIL-STD-1553A. Data bus messages consist of controller-to-remote transfers, remote-to-controller transfers and, although not presently used, remote-to-remote transfers. Data bus main frame time is a nominal 20 milliseconds, as shown in Figure 5. During this time period, the active bus controller collects data from all boxes on the data bus (via transmit commands), performs the required logic processing and computations, and outputs the revised data to all boxes on the data bus (via receive commands). For subsystems that do not require this high update rate (50 times/second), data is processed and outputed at a lower rate (25 times/second).

Bus Control

During normal operation, sole control of information transmission on the bus resides with the active bus controller, which initiates all transmissions. The primary bus controller resides within the Fire Control Computer (FCC) while the Backup Bus Controller (BBC) resides in the remote terminal unit located in the copilot's compartment.

The FCC is a general-purpose, microprogrammed, digital parallel, synchronous machine. The characteristics of the FCC are summarized in Table 1.

TABLE 1

FIRE CONTROL COMPUTER GENERAL CHARACTERISTICS

| | |
|---|---|
| Type | General purpose, micro-programmed, digital parallel, synchronous |
| Memory | 16K x 16 bits ROM (16K growth)<br>2K x 16 bits RAM |

The backup bus controller is a 16-bit, 2901A bit/slice microprogrammed processor. The characteristics of the backup bus controller are summarized in Table 2.

TABLE 2

BACKUP BUS CONTROLLER GENERAL CHARACTERISTICS

| | |
|---|---|
| Type | 2901 microprogrammed processor |
| Memory | 12K x 16 bits ROM<br>2K x 16 bits RAM |

Either the primary or backup bus controller would normally be the active controller, depending on the states of handshake discretes, a selection switch in the copilot/gunner cockpit, and activity of the busses. The block diagram of Figure 6 shows the various signals associated with the selection of primary or backup bus controller.

Both controllers monitor the busses for activity in the form of valid commands. Lights in the Copilot/Gunner (CPG) cockpit indicate whether primary or backup is in control and also cue the CPG that the backup controller has detected an inactive bus when the primary was supposed to be in control. The logic for bus controller selection is given in Table 3.

TABLE 3

BUS CONTROLLER SELECTION MATRIX

| Controller | CPG Switch | Handshake Discretes | Bus Activity |
|---|---|---|---|
| Primary to assume control | Primary | Backup not in control | None for 120 msec (programmable) or at power up |
| Backup to assume control | Primary | Primary not in control | None for 120 msec (programmable) |
| | Backup | Don't care | |

In addition to the normal bus control mode, the YAH-64 data bus system is capable of operating in a dynamic bus allocation mode. The backup bus controller was designed to recognize mode commands with the subaddress field of 00000. Subsequently, four mode codes are decoded with the 00000 code resulting in a computer interrupt in the backup controller. With appropriate computer programming the "all zeros" mode command is thus associated with the dynamic bus allocation function. The three other decoded mode codes are assigned in the present YAH-64 backup controller as shown in Table 4.

TABLE 4

MODE CODE FUNCTIONS

| Mode Code | Function |
|---|---|
| 00000 | Issues Computer Interrupt in Backup Bus Controller |
| 00001 | Clear Remote Terminal |
| 00010<br>11111 | Decoded But Not Used |
| Others | Not Decoded |

Multiplex Remote Terminal Units

There are four "standard aircraft" multiplex remote terminal unit types used in the YAH-64 data bus system.

These remote terminal units (identified as Types I, II, IIIA, and IIIB) input and output a standard assortment of bi-levels, ac and dc analog, serial digital, and synchro input/output signals to all parts of the aircraft. To fit the needs of the YAH-64 aircraft, these units contain four different input/output signal capacities as shown in Table 5.

TABLE 5

"Standard Aircraft" Multiplex Remote Terminal Unit Input/Output Capacities.

| I/O Type | Type I | Type II | Type IIIA | Type IIIB |
|---|---|---|---|---|
| 5V discrete input | 48 | | 48 | 48 |
| 5V discrete output | 56 | 8 | 56 | 56 |
| 28V discrete input | 16 | | 16 | 16 |
| 28V discrete output | 56 | 8 | 16 | 16 |
| Switch closure | 56 | | 56 | 56 |
| Synchro input | | | | 1 |
| Serial digital input | 3 | 2 | 3 | 3 |
| Serial digital output | 3 | 2 | 3 | 3 |
| Doppler | | | 1 | |
| DC analog input | 20 | 8 | 20 | 20 |
| DC analog output | 20 | 4 | 20 | 20 |
| AC analog input | 4 | | 4 | 4 |
| Precision AC analog output | 8 | | 0 | 0 |

All "standard aircraft" remote terminal units contain redundant 1553A data bus interfaces. When further redundancy is required, such as a critical input or output signal, that particular signal is wired into two separate remote terminal units.

The "standard aircraft" remote terminal unit contains sufficient built-in test circuitry to detect 95 percent of all faults (weighted by failure rate) within itself. At the LRU level, these units contain an internal test system to check input and output channels for integrity as well as power supplies and other internal circuitry.

Figure 7 and 8 show the mechanical outlines of the three types of MUX remote terminals. All three types of remote terminals have been designed for:

- Hard mounting without shock absorbers
- Convection cooling
- 100 percent plug-in modules (except power supply)
- Hand-mated connectors

It should be noted that the Type II remote terminal shown mounted in a typical pylon requires an unusual box envelope. Card commonality between all types of remote terminal units is maintained in spite of the unique packaging requirements for the Type II remote terminal unit. The cards are mounted vertically in the Type I and Type III remote terminals, and are mounted horizontally in the Type II MRTU.

Electronic Attitude Director Indicator Electronics Unit (EADI)

The EADI electronics unit receives flight symbology information via the multiplex data bus system for display on the pilot's electronic attitude indicator. Placing this unit on the data bus provides the YAH-64 helicopter with a very flexible primary flight instrument since symbols can be reformatted, added, or deleted through software modifications. Symbology is defined in the programmable symbol generator.

Symbol Generator

The symbol generator shown in Figure 9 receives mission equipment related symbology and video switching information via the data bus for display on either the pilot's or copilot's helmet mounted displays, or the heads up or down displays located in the copilot's compartment. In excess of 18 symbols can be displayed on each of these displays simultaneously.

Remote Missile Electronics

The remote missile electronics unit receives missile status and outputs missile command data via the data bus.

## Fault Detection Location Subsystem (FD/LS)

Integrated within the YAH-64 multiplex subsystem is the FD/LS. The detection and isolation of electrical/electronic LRU failures utilizes the multiplex data bus system and associated hardware for performing the various functions. The remote terminals are used for signal conditioning and data transfer. The active bus controller is used for fault processing, control, and data storage. The multifunction data entry keyboard located in the CPG compartment is used for FD/LS command initialization, and the TADS alphanumeric displays for fault location presentations. The present system fault detects and isolates over 69 replaceable units.

The copilot crew station layout showing the locations of the multifunction data entry keyboard and the TADS alphanumeric displays are shown in Figure 10. The FD/LS is capable of correctly identifying a failed LRU in the air or on the ground for at least 95 percent of the failures monitored, weighted by failure rates. In addition, the system false alarm rate does not exceed 2 percent.

The active bus controller operating from data received over the multiplex data bus, using equipment status stored in memory, processes programmed algorithms as required to determine the operational status of the monitored LRU. The bus controller outputs the required information to the symbol generator which generates alphanumeric messages for display on the TADS alphanumeric display.

When in flight, the FD/LS continually and automatically monitors the operational status of the YAH-64 electronic/electrical subsystems. If a fault occurs, the aircrew is alerted so appropriate action can be taken. Major subsystems which are so monitored include the missile subsystem, rocket subsystem, gun, automatic stabilization equipment, pilot night vision, integrated helmet mounted sight and display, doppler, target acquisition and designation system, symbol generator, EADI, engine instruments, auxiliary power unit and electrical subsystem.

Through special keyboard entries, either flight or maintenance personnel can command a particular subsystem fault detection/location test routine. At the completion of the test, the results are displayed on the TADS alphanumeric display for evaluation. In addition, in the maintenance mode, a complete aircraft end-to-end fault detection/location test can be commanded through a maintenance keyboard entry. This test will check all systems connected to FD/LS, and will display all failed units on the TADS alphanumeric display.

Summary

Presently the YAH-64 multiplex system described in this paper is going through the initial integration phase of bench checkout and software development. Bus interfacing equipment including the FCC, BBC, RHE, SG, and all MRTUs have been successfully integrated with both primary and backup bus controllers in command. Non-bus hardware interfaces of the complete Hellfire system and IHADSS have been checked out. Software development is about half complete with approximately 6000 words written and integrated. The present schedule calls for a flight test of a partial MUX system in October 1978, with bench testing of the total system to be complete by January 1979. First flight of the total MUX system is scheduled for April 1979.

## DEFINITIONS

| | | |
|---|---|---|
| BBC | - | Backup Bus Controller |
| EADI | - | Electronic Attitude Director Indicator |
| FCC | - | Fire Control Computer |
| FD/LS | - | Fault Detection/Location Subsystem |
| HARS | - | Heading and Attitude Reference System |
| HMMS | - | Hellfire Module Missile |
| IHADSS | - | Integrated Helmet and Display Sight Subsystem |
| ORT | - | Optical Relay Tube |
| PNVS | - | Pilot Night Vision System |
| RHE | - | Remote Hellfire Electronics |
| SG | - | Symbol Generator |
| TADS | - | Target Acquisition and Designation System |



Figure 1. Hughes Helicopters YAH-64

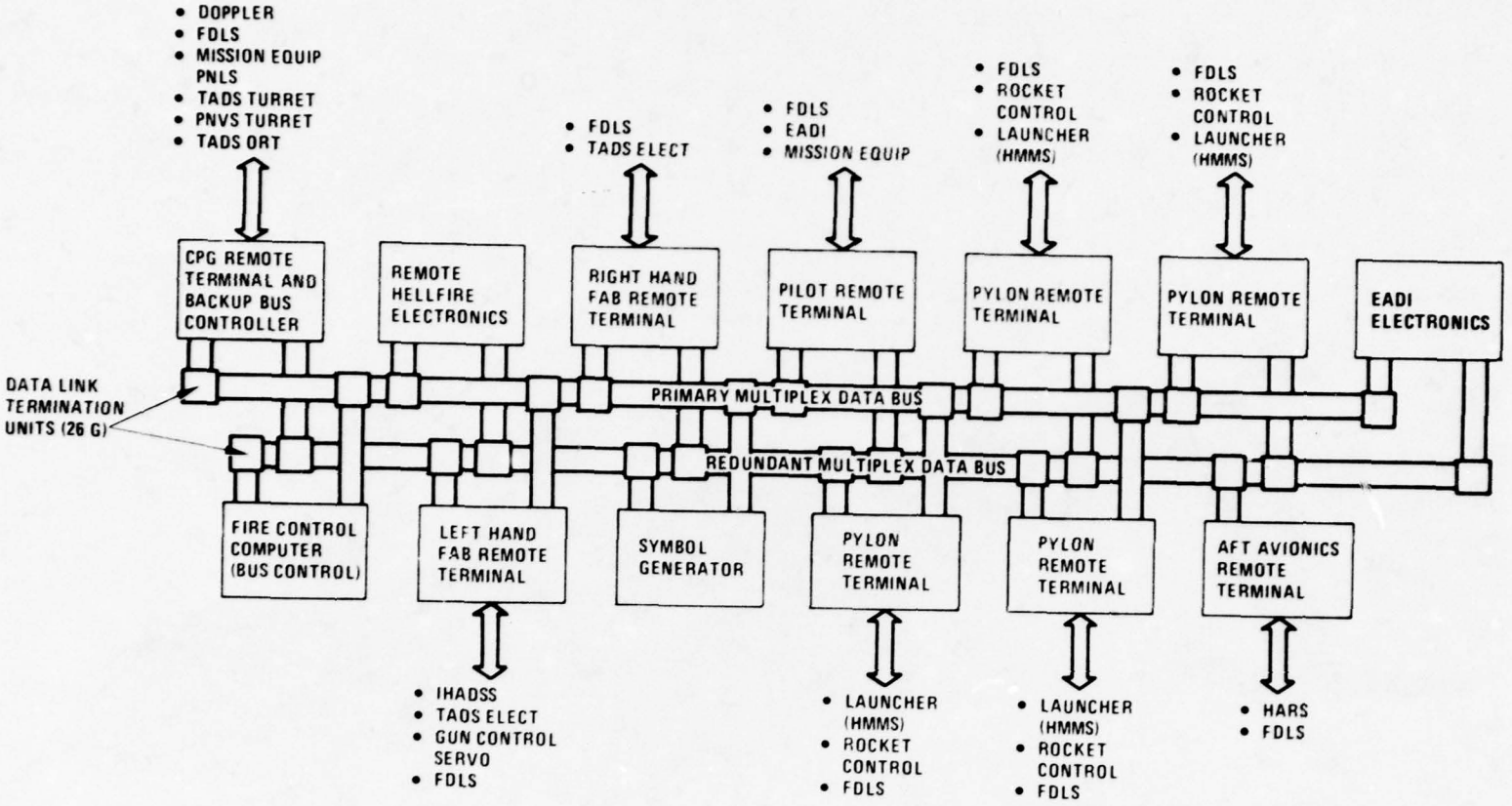



Figure 2. YAH-64 Multiplex System Block Diagram

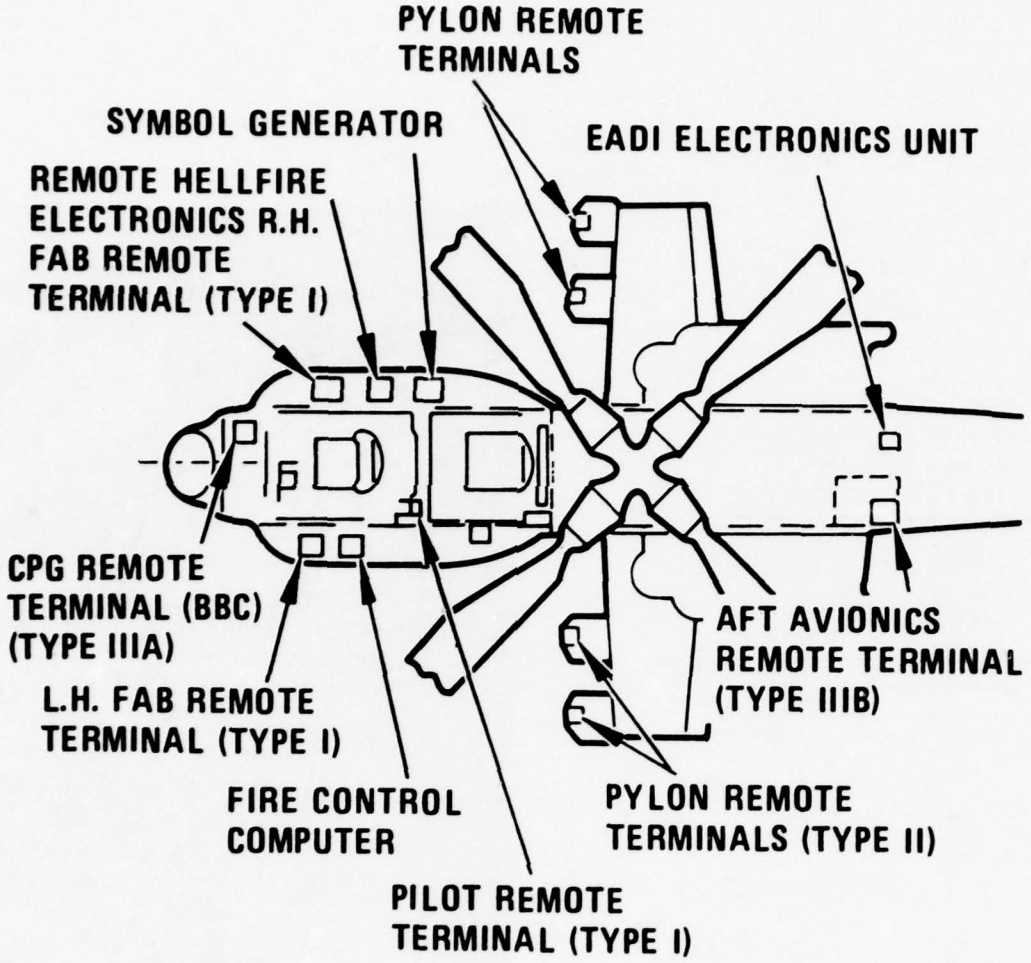
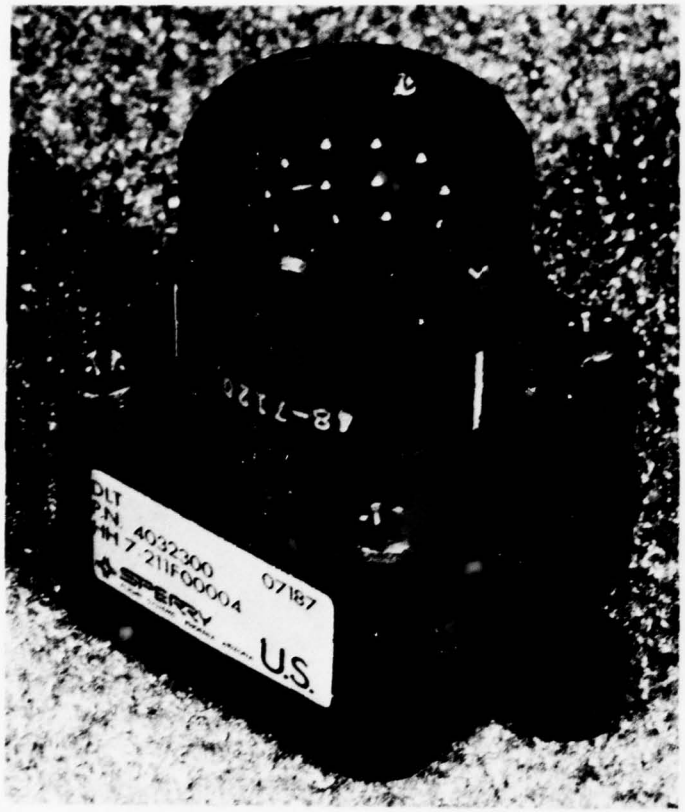



Figure 3. YAH-64 Multiplex System Unit Locations



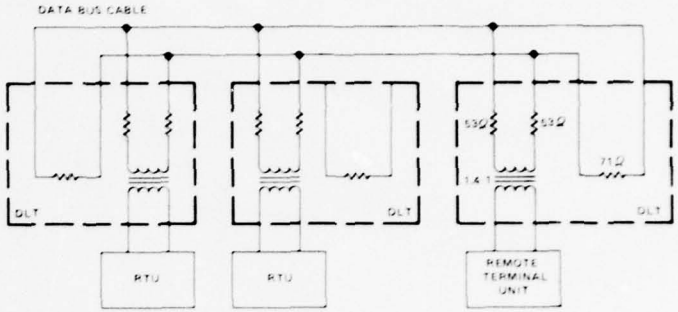




Figure 4. Data Bus Cables

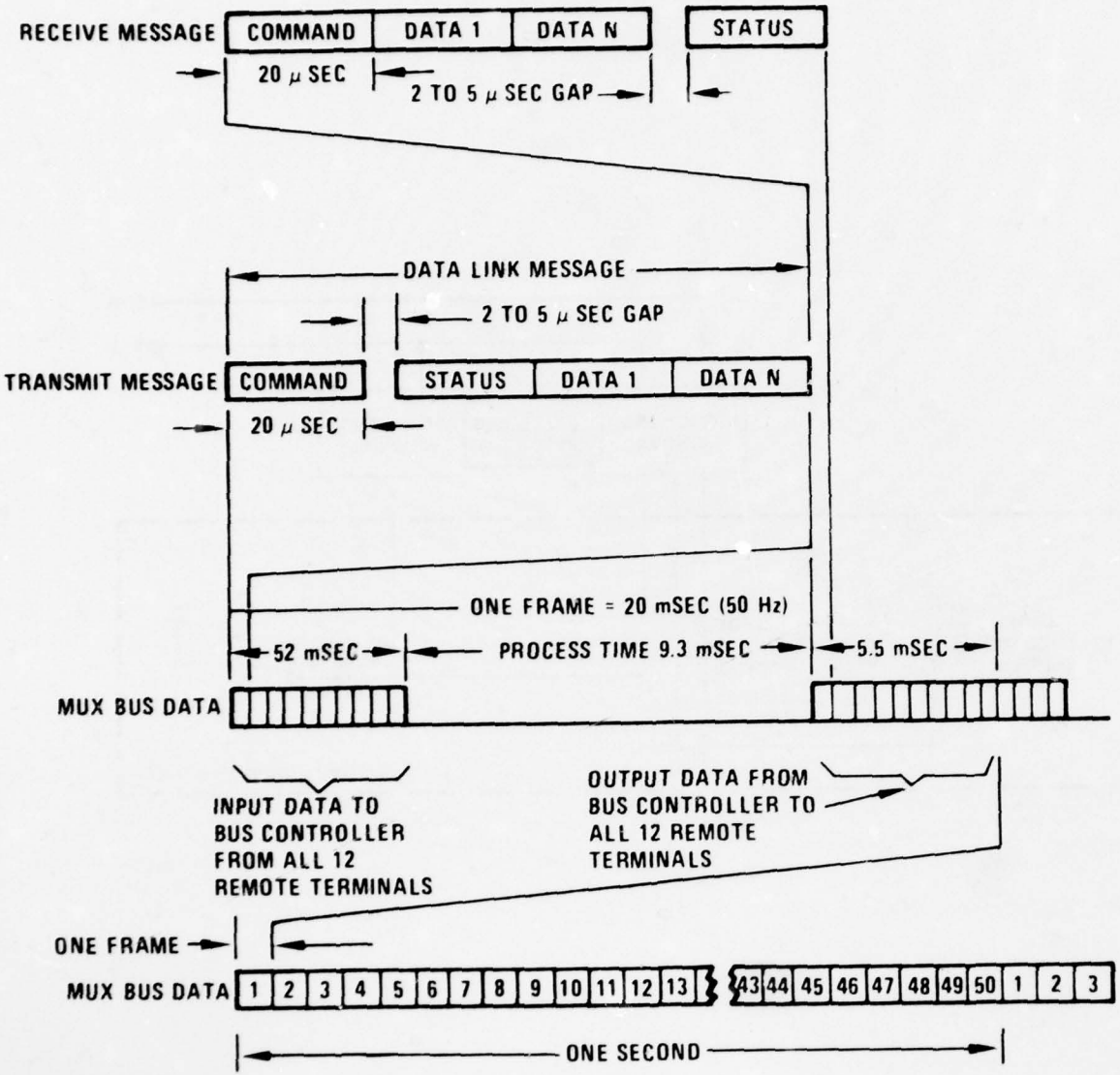



Figure 5. Multiplex Data Bus Frame-Time Formats

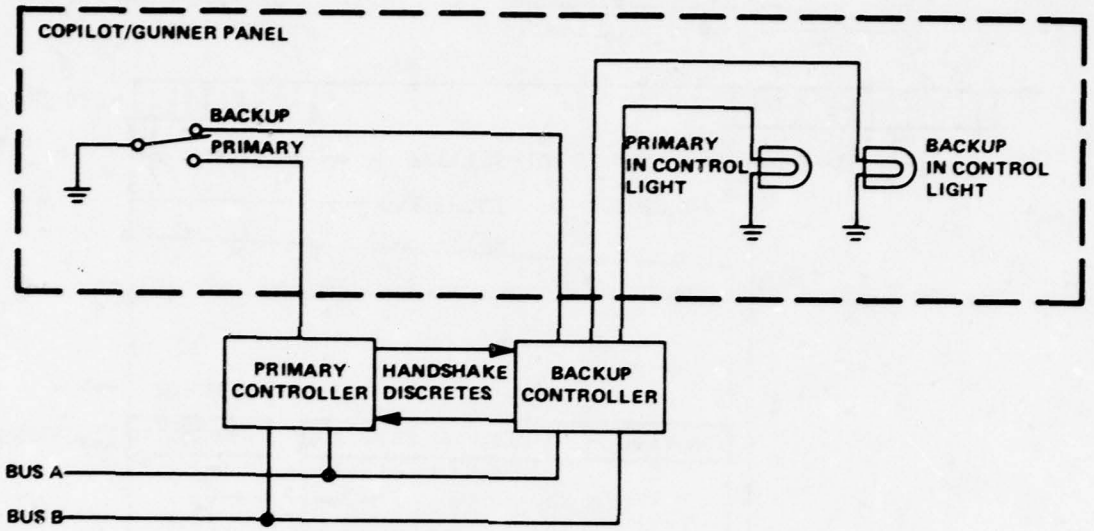



Figure 6. Bus Controller Handshake Technique

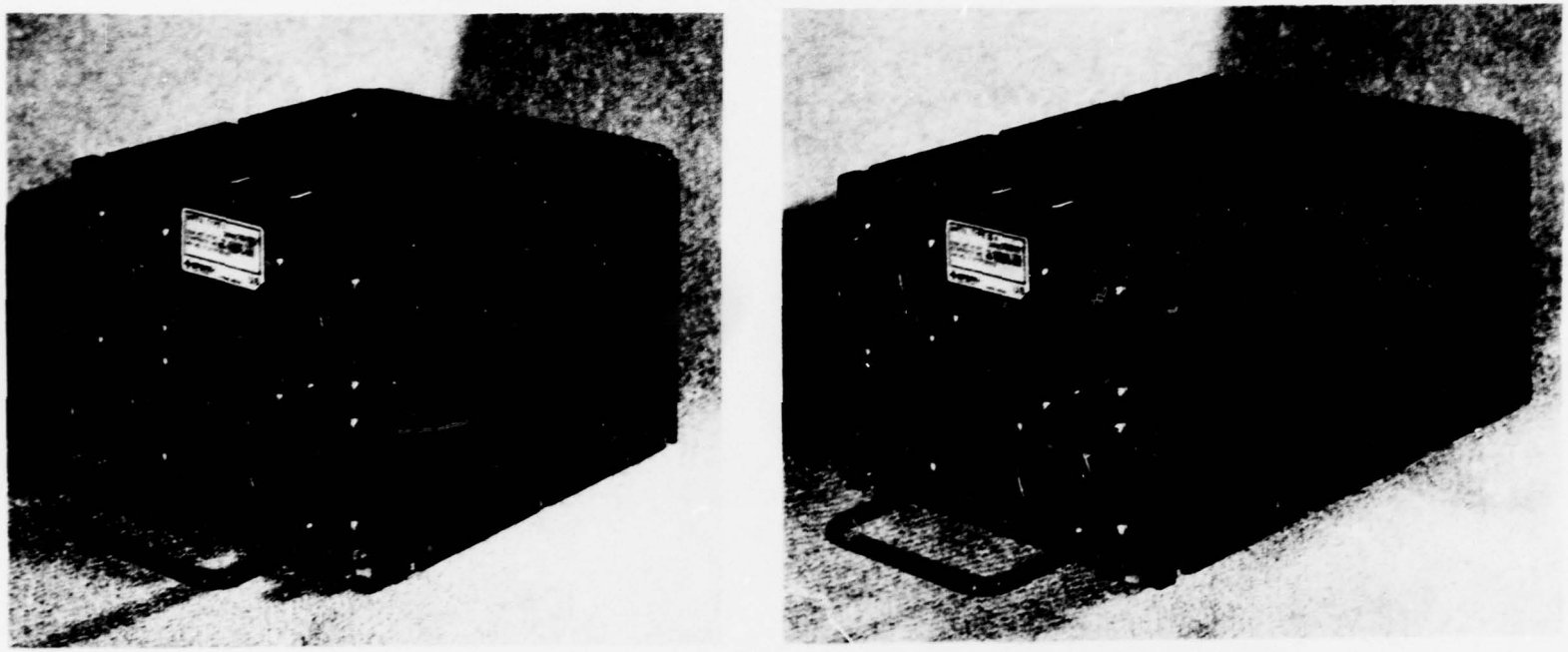
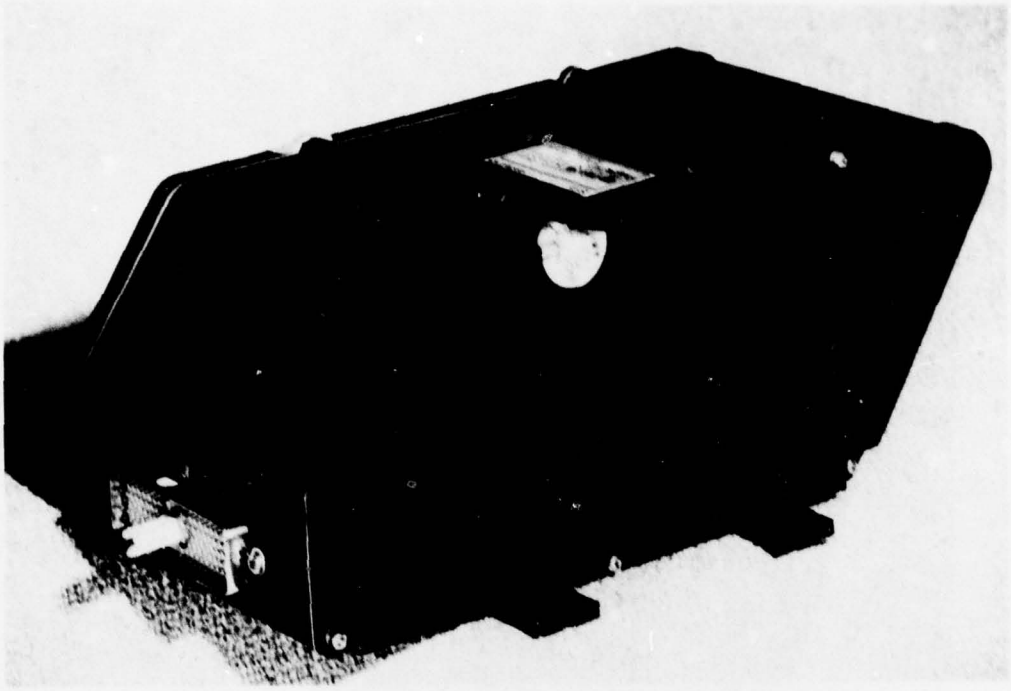


Figure 7. Type I and III Remote Terminal Units



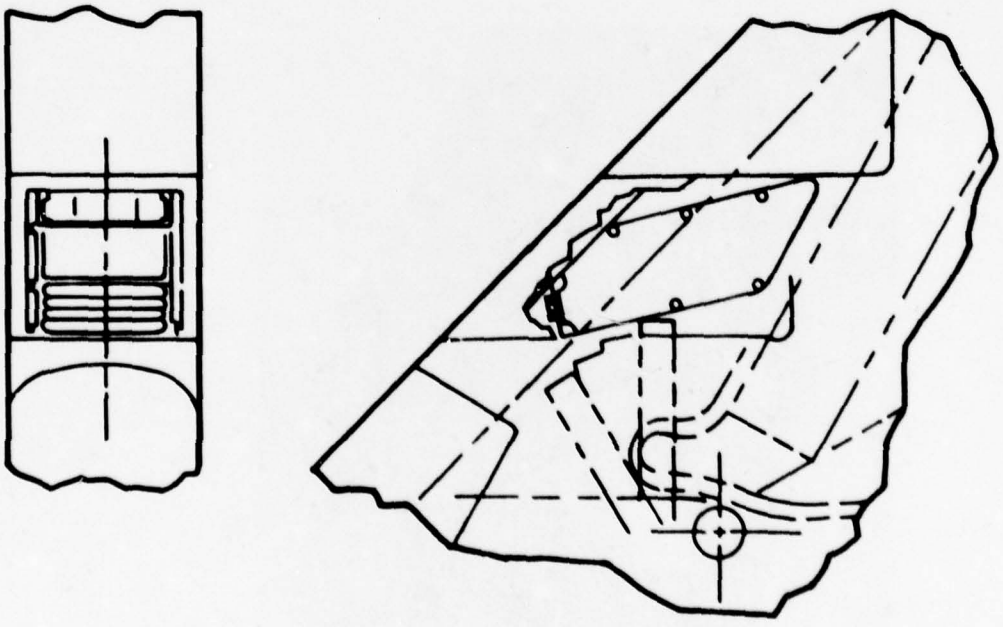


Figure 8. Type II Remote Terminal and Pylon Mounting

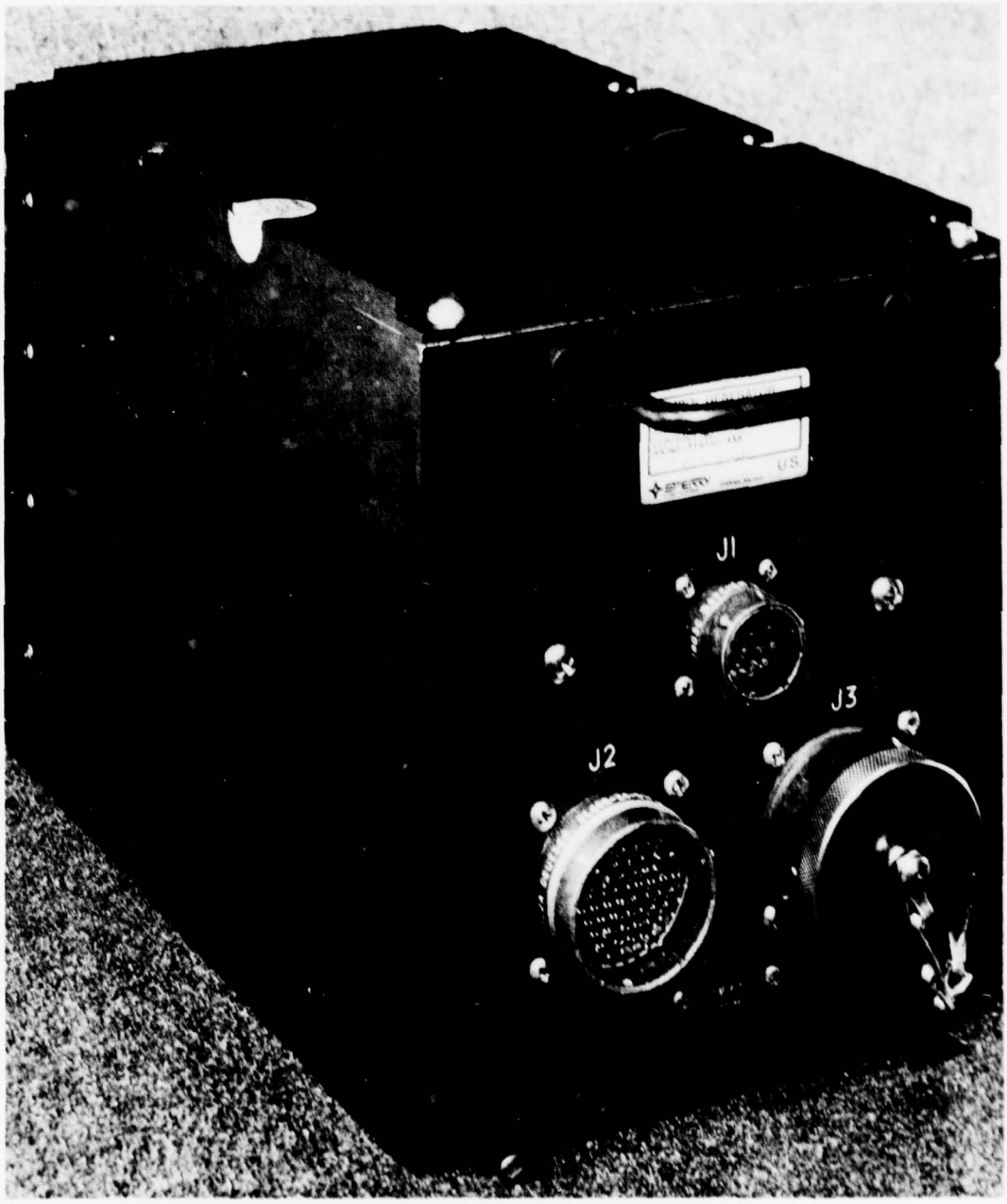



Figure 9. Symbol Generator

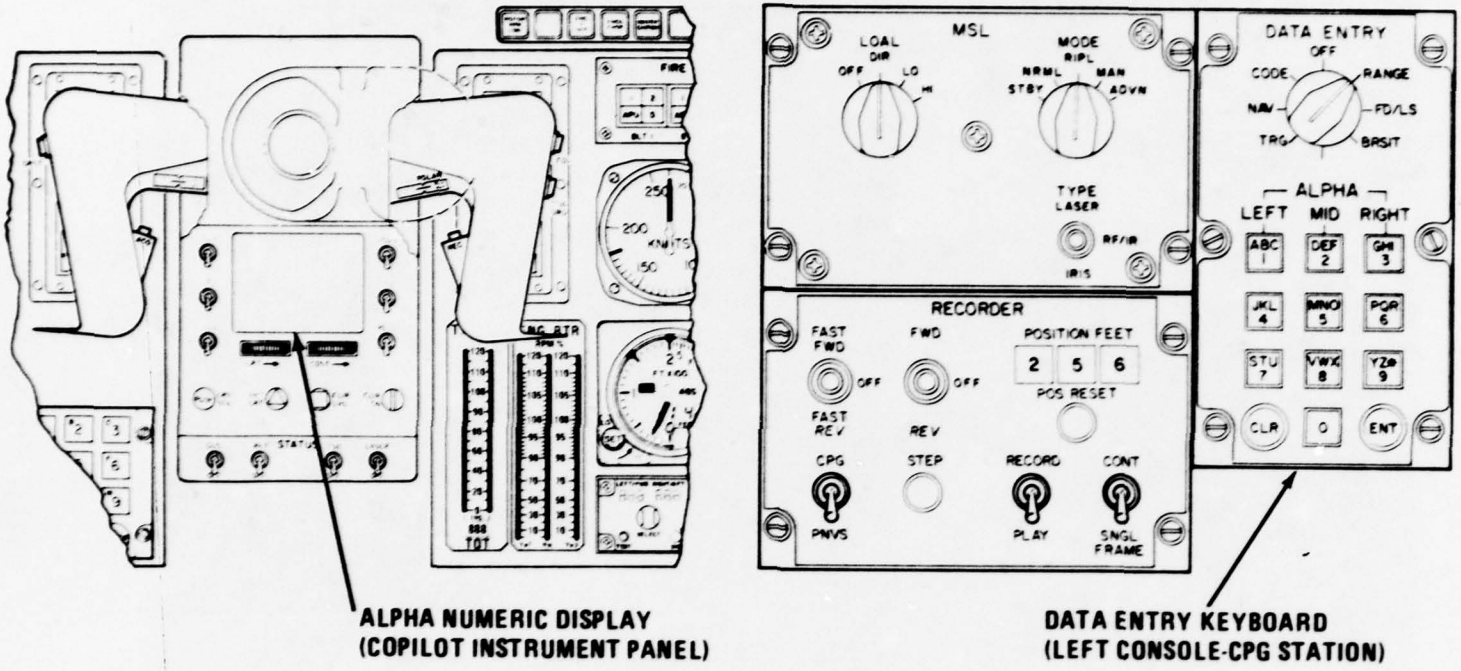



Figure 10. Fault Detection/Location Subsystem Controls and Displays

# EVOLUTION OF A MULTIPLEX SYSTEM

by

George Cotter
Larry Driscoll
Howard Sherman

Grumman Aerospace Corp.
Bethpage, N.Y., 11714

ABSTRACT

This is a review of the evolution of multiplexed systems which use distributed microprocessors. Two previous papers[1,2] discussed the organization and application of "SMART Terminals". This paper covers actual systems applications, hardware designs and their operational benefits.

Grumman has been developing systems using microprocessors with multiplexing over the past 5 years. We started with an 8080 design which was used in a solid state electric system (SOSTEL) lab development system. This was followed by a design which used 3000 microprocessor chips in a 16 bit configuration for general purpose remote multiplex terminals, again for lab use. Both of these designs used discrete circuits for the multiplex front end. Our next design was an eight bit version of the 3000 series microprocessor and used hybrid receivers and transmitters and an LSI for the Manchester front end. This design is incorporated into the Pilot's Control Panel delivered to Eglin Air Force Base for their advance development store management system. Our latest multiplex front end design operates with our eight

1. Avionics Multiplexing with Smart Terminals Oct 75
2. Avionics Multiplexing with Smart Terminals (An Update) Nov 76

bit 3000 microprocessor or a discrete logic unit and uses hybrid transceivers and two LSI's, one for the Manchester front end and the other for 1553A message control. This design has the dual bus interface on a 4.25" by 4.5" card and is used in our Integrated Avionics Control Set (IACS) and the hydrofoil engineering monitoring and control system (EMCS).

The important aspect of this design is not the bus interface or microprocessor designs. What is important is the flexibility provided to the system designers and the human factors engineers. In the following paragraphs we will discuss the hardware and system designs as we have seen them to date.

## HISTORY

Decades ago, an airborne electronic system consisted of a few relatively simple pieces of equipment operating independently. Today, aircraft rely on a greater number of sophisticated multi-purpose interdependent black boxes for protection and completion of their missions. The main challenge to the avionics industry for decades has been the shift to greater reliance on electronics to perform in an environment which restricts size, weight and power consumption. The undertaking has been made even more difficult by the need to add more functions and capabilities into airborne platforms as the technology advances. Microminiaturization has helped decrease equipment size, weight and power consumption greatly. It has also helped solve the demand to add more electronic capability. However, the added capability has increased the difficulty of integrating the system components by establishing additional interfaces. The addition of more equipments has caused a reduction in cockpit space and an increase in pilot workload required to exercise them. The operational impact has become so severe that pilots are now describing their tasks as - overloaded, overburdened, error producing environment, etc. So, emphasis has shifted to generating systems that produce a reduced amount of more meaningful information for the crew while integrating hardware functions to save equipment space in the cockpit. The multiplex approach satisfies this need completely. Multiplexing techniques, along with microprocessor controlled remote terminals, allows centralized cockpit control of several subsystems through a single keyboard and display. The basic electronics associated with these subsystems can be remoted in available non-critical space in the vehicle.

In addition, we can significantly relieve the pilot workload by performing for him the mental integrations he formerly required in using stand-alone equipments. Helicopter pilots for example, currently required to mentally integrate torque, outside air temperature and engine RPM can now receive a direct readout of percent power - the information he really needs.

MUX implementation also allows us to make redundant functions immediately available with a lot less hardware than required by stand-alone equipments. And - we can get a switchover which is performed smoothly and automatically so that the crew can continue to operate without a break in continuity. At some reasonable break-point, we inform the crew they are on a back-up system.

## OPERATIONAL CONSIDERATIONS

Advantages gained from using a centralized cockpit system are obvious. There are, however, pitfalls to be avoided if the implementation is to be successful. Most of these pitfalls are eliminated through good human factors design. The systems we are addressing are being produced for use by a human operator. It is therefore mandatory to insure that the operator can easily perform his tasks with these equipments. If he cannot - he'll make mistakes. And when he makes errors, he will soon stop using the equipment even if he has to break it to make that possible. During our Integrated Avionics Control Set (IACS) program, we frequently evaluated the amount and quality of information provided to the operator. IACS equipment centralizes control in the cockpit for a number of navigation and communication equipments. Although the configuration of the IACS system permitted display of a large amount of information to the crew, an analysis established the final information requirement. This information was limited to only the minimum required for the crew to perform the task. Too much information can be confusing and will be unavailable in a quick scan.

The IACS primary display equipment is shown in Figure 1. The display on the Primary Control Panel is limited to five rows of eight characters. This display has proved more than adequate for presenting IACS information and was readily useable for incorporating doppler information as well. The doppler was a growth feature. The display is red incandescent to meet a program requirement for visibility in high ambient light. The incandescent light is piped through fiber optics

to form segment readouts - which are encased in a stycast block. The face cannot be dirted, smudged or broken to cause display degradation as is the case with glass faces and filters.

Another major design concern is the keyboard switch pushbutton force. These new systems require extensive interaction between the operator and the equipments through a keyboard. This is a new task for a flight crewman and he approaches it with skepticism. Unlike the average portable computer keyboard, establishing forces for a keyboard on flight vehicles assumes a role of major importance. The operator wears flight gloves aboard our military flight vehicles. The flight vehicle is rarely as stable as the desk top. On the IACS program we found the bubble switch to be inadequate for our purposes and have gone to a miniature switch of leaf design. During our testing it became obvious that you could meet force requirements but not provide reasonable feedback to a point where the operator found the whole system unacceptable. With this switch we are able to obtain our desired forces while maintaining a positive feedback to the operator.

One other error contribution should be emphasized. When controlling a large number of equipments via software, it is mandatory to establish the operating philosophy early. This philosophy must be consistent with the past experience of the crewman. If the requirements for operating are inconsistent with the operator's past experiences on those or similar equipments, we have built in an error producing situation. You cannot, for example, make the operator "turn pages" from top to bottom if he is used to turning from bottom to top. Operations should be minimized wherever possible. Our IACS system utilizes essentially the same operating philosophy for its centralized control of remoted equipments as the individual equipments had originally. In simulations of military missions our operators made no mistakes with the equipment after only an hour of training.

## REDUNDANCY

Multiplexing provides a means of transferring data and commands from multiple units over the same transmission line(s). Multiple sources and destinations for the data and commands are accommodated within the system architecture. When the associated subsystem units utilize microprocessors the degree of system backup or number of modes for degraded operation is limited technically only by the penalty of

additional firmware, size, weight, and power. In addition, the procurement costs, both developmental and per unit price must be considered.

Determination of system redundancy can be accomplished by assigning a priority to each system function and providing back-up operation based on the subsystems (or multiplex units), potential failure rate, impact on crew workload, etc. as related to the mission requirements. An example of this would be two interactive control panels; one assigned to stores management; the other to Comm/Nav control. Both panels could perform both functions if required and provide redundancy with only the addition of firmware. Using this approach, a single keyboard can be used for all subsystem data entry requirements as long as there is a means of indicating the subsystem being addressed. A second keyboard then provides full redundancy for all system keyboard requirements.

## GROWTH AND FLEXIBILITY

The 1553A multiplex standard provides for two basic control modes, command/response and dynamic bus allocation.

In the basic command response operation remote terminals receive and transmit data only when commanded to do so by the bus controller. The bus controller would tend to be a central processor which would then function to control the flow of data among all system elements. In addition, a back up controller is required which is capable of recognizing faulty primary bus control operation and taking over the system control function. The ability of the bus controller to minimize data bus traffic when operating in conjunction with "intelligent" subsystems becomes a significant controller design criteria. Should an associated subsystem perform local processing functions and transmit data only on an "as required" basis this additional task must now be resident in the bus controller for total system compatability. Under the command/response multiplex control, any changes in associated subsystems will require a change in the primary and secondary controller software (firmware).

In dynamic bus allocation designated terminals within the system are offered control of the data bus. This approach provides a powerful tool for a system in which growth and changes can be accommodated with a minimal impact. As subsystem changes occur the impact is limited to the particular unit or units associated with the change. In

addition, the design criteria and requirements for the bus controller become minimal since the primary function is periodically to transfer bus control from one subsystem to another. The back-up controller must still be capable of recognizing faulty primary bus control operation and taking over the system control functions.

## SYSTEM APPLICATIONS

The Integrated Avionics Control Set (IACS) AN/ASQ-166 is an integrated Comm/Nav control system. This is an application where the benefits are primarily operational. Valuable cockpit panel space is opened up, redundant control capability is provided, centralized controls reduce work load and redundancy permits graceful degradation in case of failures.

The block diagram in Figure 2 shows one possible system configuration. All five units use the BIU card (Figure 3) and two units incorporate the microprocessor module (Figure 4). The primary control unit also contains an eraseable NDRO memory card which stores all presets and last operating modes.

Two microprocessors are used in IACS to provide redundancy for bus control and system operation. When the IACS operates on a stand-alone basis any unit can fail and some Comm/Nav capability will still be available. In fully integrated 1553A systems, IACS operates as a subsystem utilizing dynamic bus control. This approach minimizes the impact on the system bus controller software. IACS could also operate in a fully remote configuration using terminal to terminal transfers. This approach places the burden on the system bus controller to initiate all data transfers, which results in an increase in bus overhead and controller software complexity.

Operationally, the two microprocessors guarantee that as long as one control panel is operational full system control is available. The system loses control capability only when a control unit is lost. Then only the units interfaced by that control unit are lost. It can be seen that as more equipment is interfaced directly to the bus as the ARN-128 is, the greater the benefits of a multiplexed system.

The Primary Control Indicator (PCI) provides full system control via dedicated switches and interactive display/switch function. All data shown on the fiber optic alpha-numeric displays is under control of the microprocessors.

The switches on the panel are also read by the microprocessors and are interpretted based on what is on the display. This approach permits modification and addition of control functions with only a firmware change. Growth of this type is provided for in the present hardware.

The Secondary Control Indicator operates using either microprocessor for controlling the single line display and interpretting the control switches. Since the same firmware is used for both primary and secondary panels the control capability is the same. The only limitation is operational due to the limited display on the secondary control.

The Status indicator is a repeater which will display whatever is sent to it via the 1553A bus. In IACS this unit is used to indicate channel, frequency, and secure status of the selected transmitter.

The control units provide the function of making todays Comm/Nav equipment compatible with a 1553A system. As new avionics are developed with 1553A interfaces, units such as these should gradually disappear.

Thanks to multiplexing and microprocessors, the IACS units can be used in different combinations to meet a system requirement. In the past we would have had to change the system around to accommodate the hardware. By assigning the terminal addresses we can design a system with two primary control indicators, two status indicators, a secondary control indicator or any other combination of units. With a software change the control function could be a store management system, a solid state electric system, or even an engine monitor and control system.

## HYDROFOIL ENGINE MONITOR AND CONTROL SYSTEM (EMCS)

In this application we are using the same building block as in IACS to provide an EMCS (Figure 5) which is fault tolerant and not burdened by the penalties of a fully redundant hardwired system.

The EMCS is made up of six "intelligent" multiplex terminals each capable of monitoring and evaluating input signals representing subsystem operational parameters. These six terminals provide all of the control and monitoring functions required to operate the basic hydrofoil boat control subsystems.

Control and display functions operate the same as IACS using interactive alphanumeric displays with multifunction pushbutton switches on each side. The engineering operating station (EOS) has two sets of interactive displays; one general purpose and one dedicated. The general purpose display provides for selection of all parameters monitored by the system including those on the dedicated displays. The multifunction switches on each side permit control of any function which is programmed onto the display. A dedicated control/display is used to present system parameters which are considered as critical and should be displayed at all times. Either display can be used as a backup for the other display in case of a failure.

With each control/display being driven off of different terminals a single failure cannot wipe out all control/display capability. This partitioning of functions to limit the amount of system degradation due to a failure is also incorporated into the selection of monitor and control parameters for each terminal. Any control or parameter considered to be critical is routed through two or more terminals. This provides a level of redundancy which would be very costly to accomplish with a hard wired system. Since each terminal handles its own processing a true partitioning of the system software is accomplished. This also reduces the traffic on the bus for now only processed data required for display or control is on the bus while data evaluation is performed at the terminal. The processor in each terminal also provides benefits where multiple failures have occurred. The terminals can be programmed to output a predefined set of control functions or a control function based on data being monitored within the terminal. In this way if both bus controllers are lost and there is no other terminal programmed as a backup controller, a safe control situation can still be maintained.

## HARDWARE

We have developed a group of building blocks for use in designing multiplex systems. Some of these are the bus interface unit (BIU), microprocessor module, control card for "dumb" terminals, standard I/O cards, and high efficiency (75%) power supply modules. The BIU and the microprocessor module are described in the following paragraphs.

Bus Interface Unit (BIU) - Figure 3 is a picture of the BIU card which provides a dual 1553A bus interface. The block diagram in Figure 6 shows the functions incorporated in the

BIU. The transceivers are hybrids, the encoder/decoder and buffer are forty pin LSI's, and discrete circuits are used for watchdog timer and bus switching. Most of this is very similar to other designs presently in use with the exception of the buffer LSI. This device is designed to provide all basic 1553A message control while providing a simple interface to either a microprocessor or fixed logic terminal design.

The buffer's primary function is to provide the serial to/from parallel conversions, the controls, interrupts, and status information for the user. It is designed as a 40 pin LSI device and maintains the independent transmit/receive paths of the encoder/decoder. Data transfer functions are provided by a 16 bit transmit and a 16 bit receive buffer storage. These are organized in 8 bit bytes to accept or deliver data to the 8 bit tri-state bus in response to take and read signals from the user. Interrupts and controls are generated in response to command word reception of the command type (transmit, receive, broadcast). Mode controls provide the flexibility to initialize the unit, initiate or respond with transmissions, and provide the flexibility for the unit to operate as either a remote terminal responding to commands or as a bus controller initiating commands. Status information is provided as tri-state (or discretes) outputs which alert the user of unique polling offer word fields, data transfer alerts, and critical operating conditions or modes. The unit has several unique and automatic functions such as a message complete function. This operates to alert the user of the thirty-third word transmitted in the case of a bus controller. In addition, as a remote unit, it is generated in response to the correct number of data words received or the correct number of words transmitted and initiates and/or stops transmission. Internal idle line resets cause the unit to revert to the receive mode in the event of message failures.

Microprocessor - The most important part of a flexible design is the microprocessor and its marriage to the BIU. Figures 4 and 7 are a picture and block diagram respectively of an eight bit microprogrammed microprocessor. The CPE's and look-ahead carry on one card, and the interrupt control, micro control, 512 word micro-memory, and pipe-line on a second card which plugs into the CPE card. The microprocessor instruction set has been designed specifically for multiplex terminal applications. There are special instructions for bus control, bus interface and input/output data handling. Eight priority interrupts are provided with the two highest reserved for the transmit and receive interrupts from the BIU buffer.

The processor design provides for addressing up to 64k words of memory using a memory extension register under control of a micro bit.

## CONCLUSION

Multiplex technology bridges the gap between hardware and system development. We can now satisfy operational requirements in a realizable and controllable manner which allows the use of current and near future avionics hardware. Our experience with the systems presented has confirmed that the basic building block of multiplex technology will be the cornerstone of future systems.

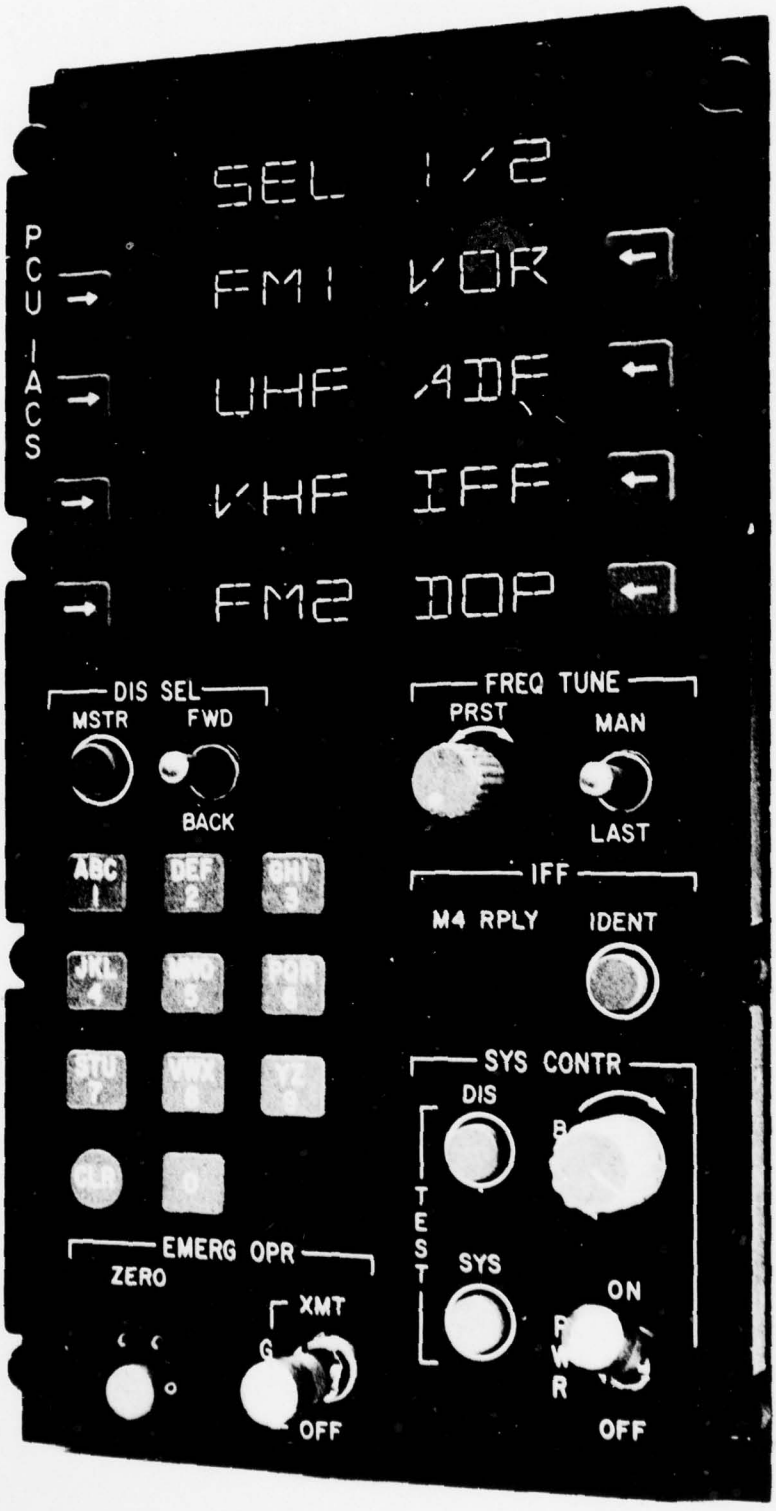



FIGURE 1. IACS PRIMARY CONTROL INDICATOR

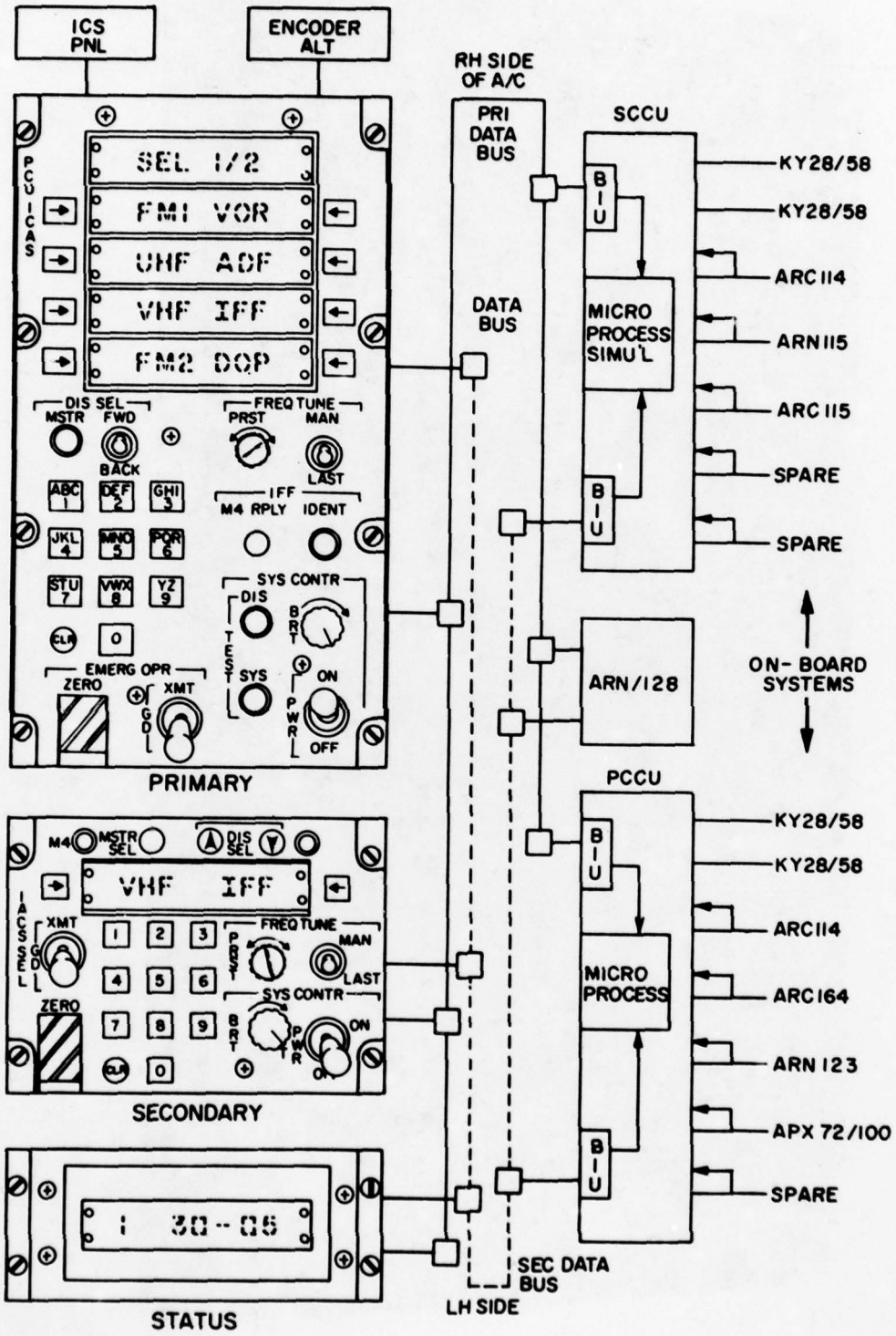



FIGURE 2. IACS SYSTEM

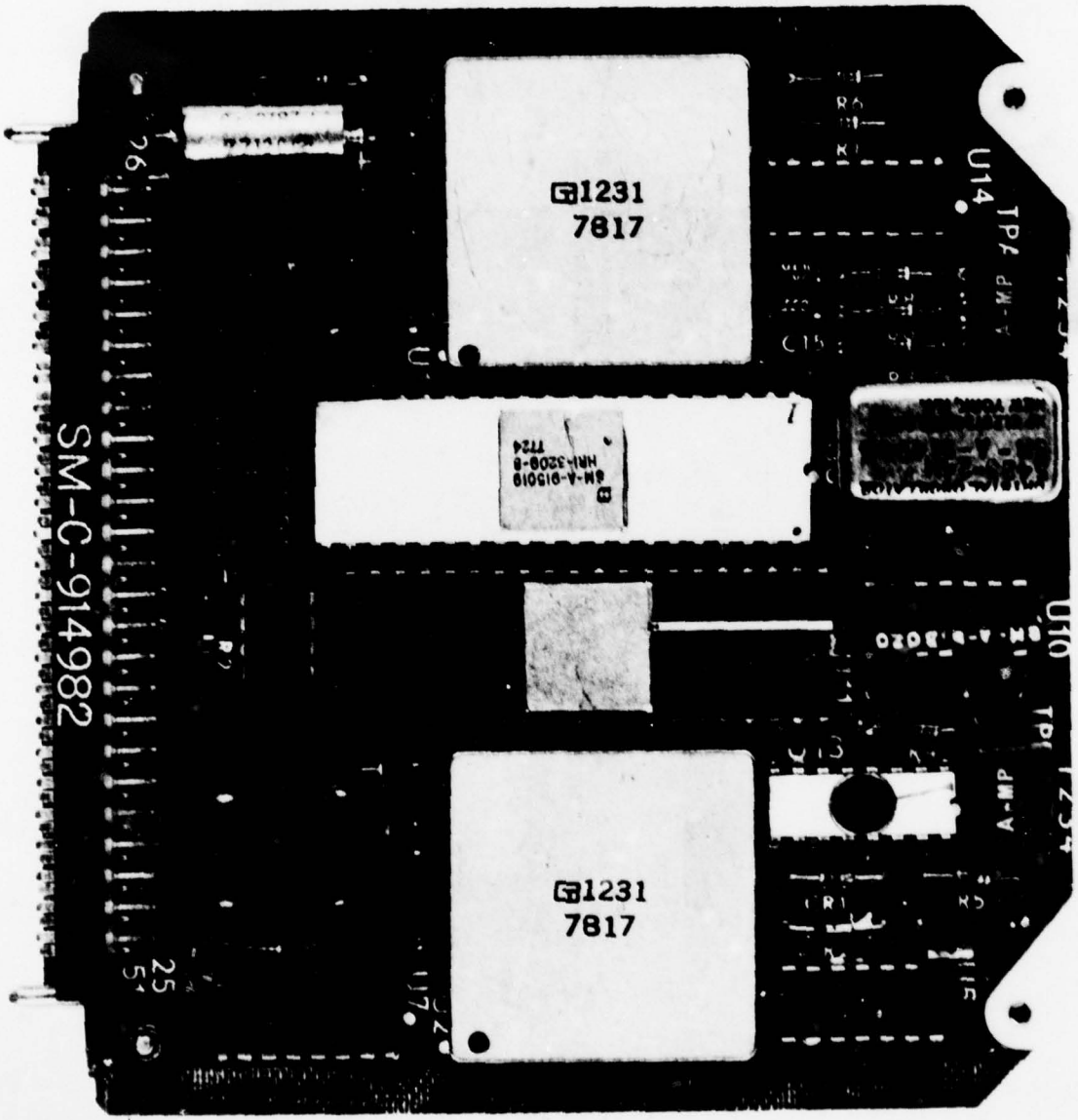
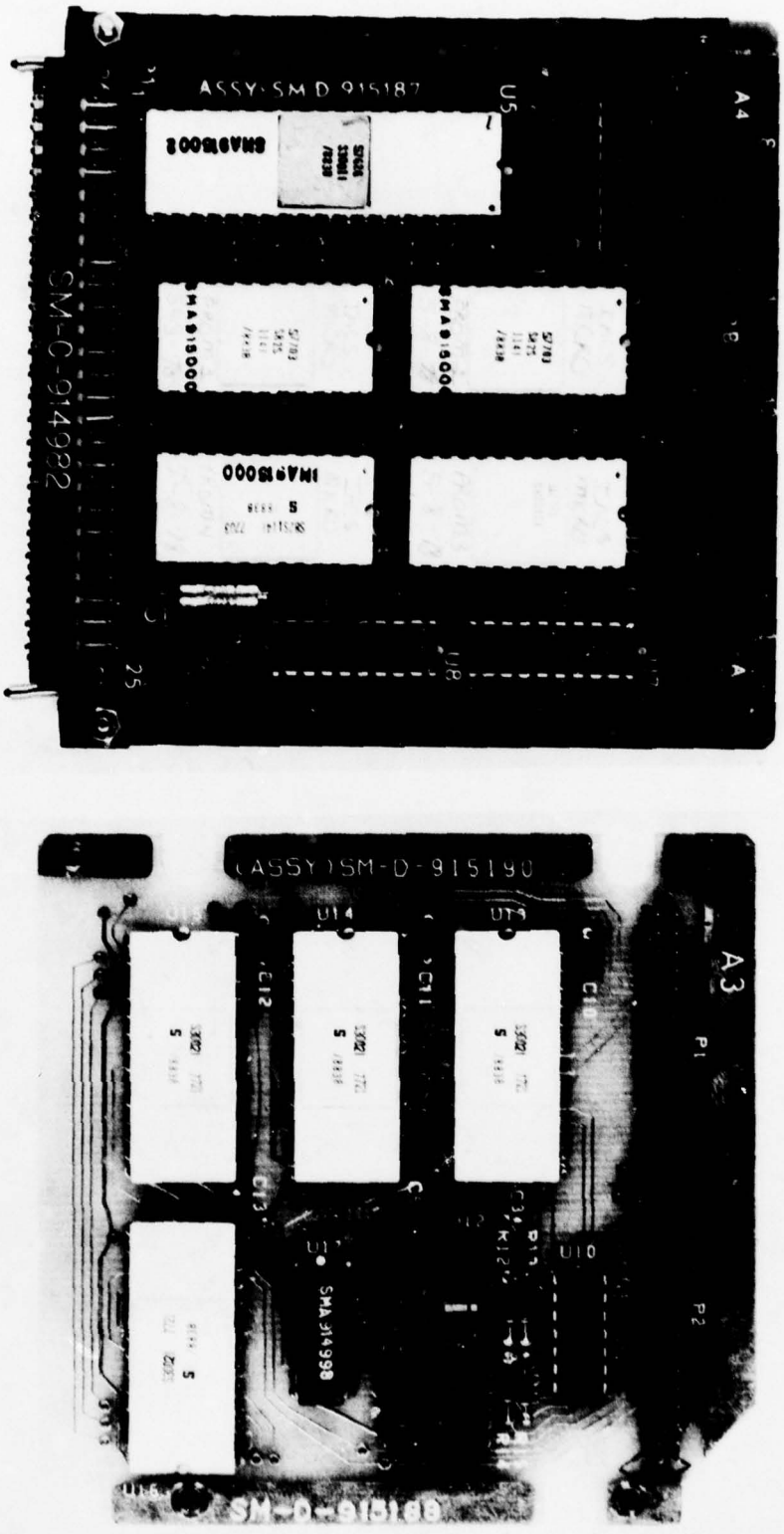



FIGURE 3. BUS INTERFACE UNIT


ASSY-SM D 915187
SM-D-914922
A4
A3
P1
P2
U5
U8
U14
U19
U10
U17
SMA91000
SMA91000
SMA91000

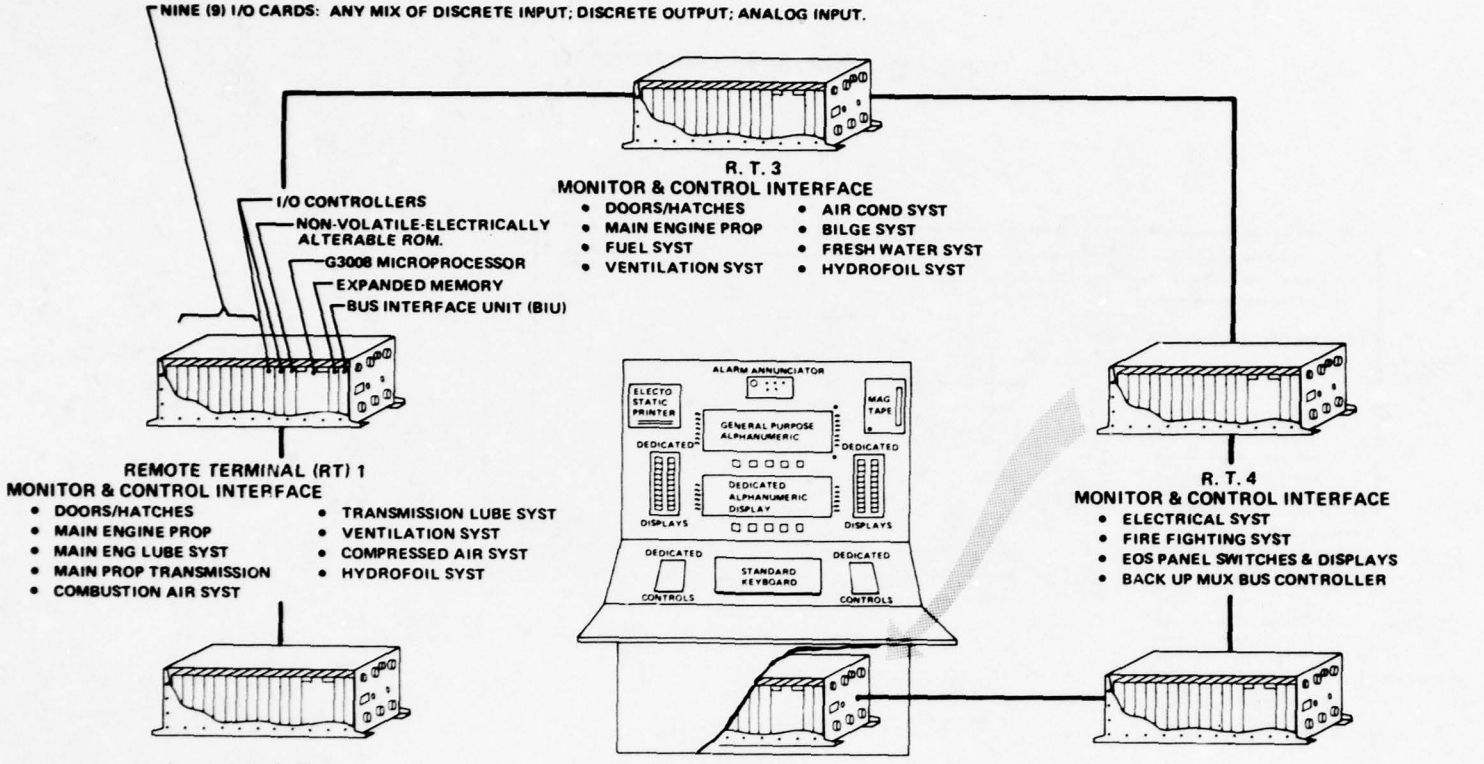



FIGURE 5. ENGINEERING MONITORING & CONTROL SYSTEM (EMCS)

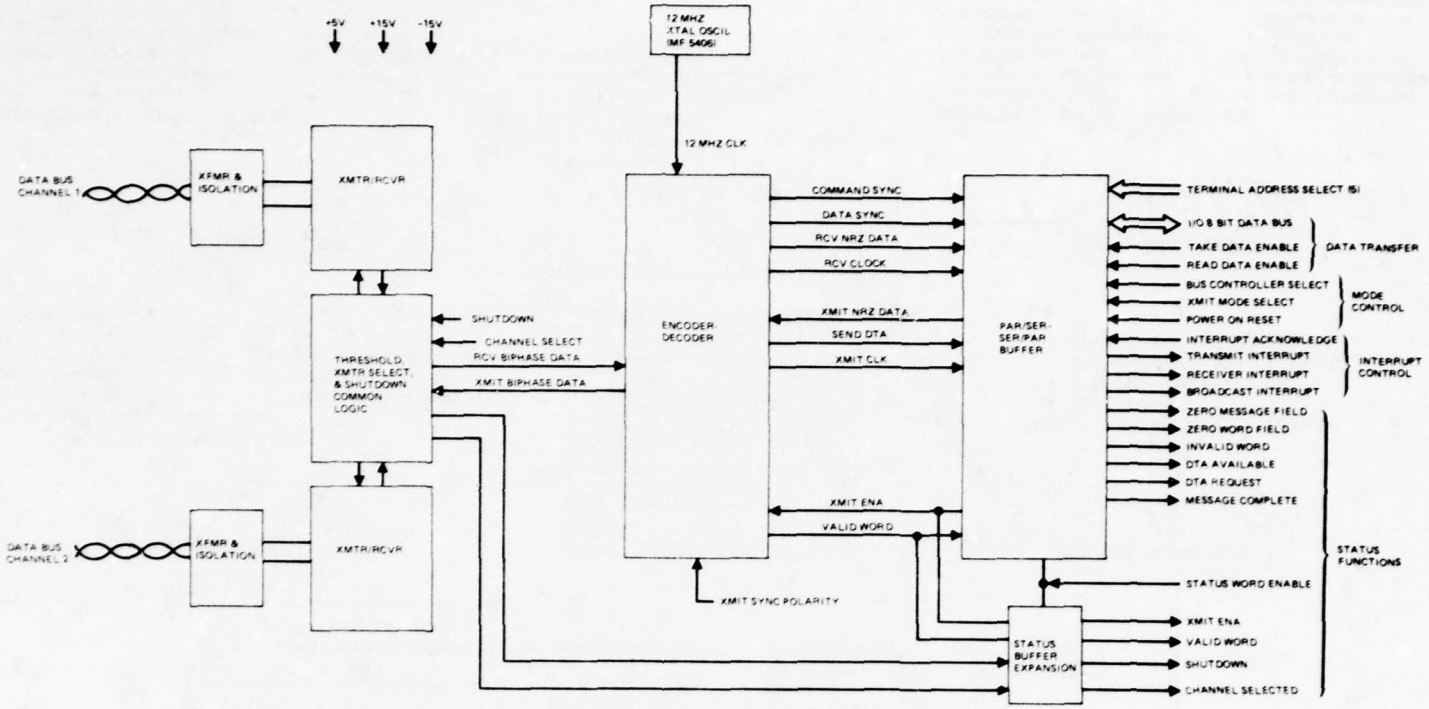



**FIGURE 6. DATA BUS INTERFACE CARD.**

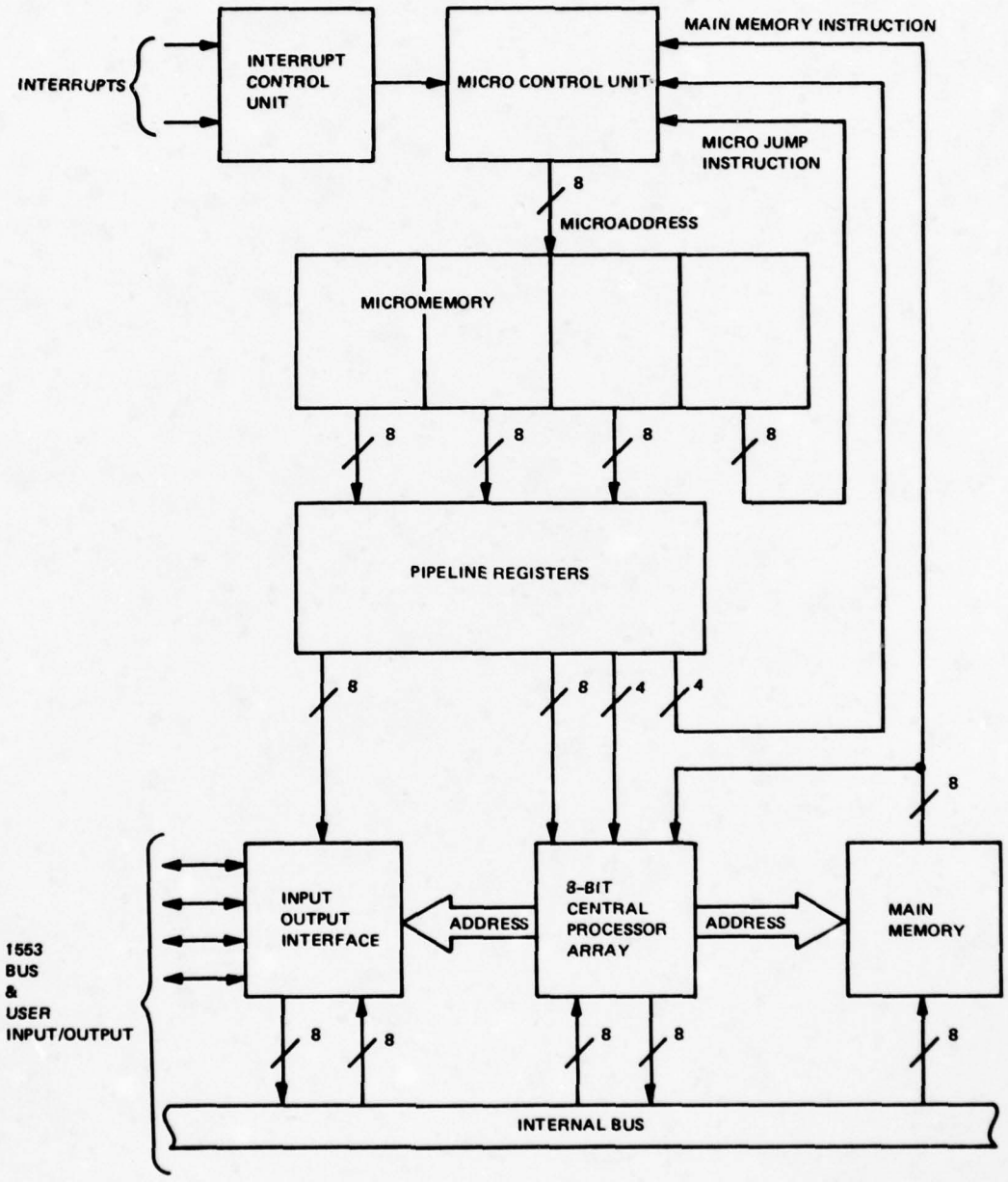



**FIGURE 7. MICROPROCESSOR CARD**

# INVESTIGATION OF DATA SYSTEMS ARCHITECTURES FOR STORES MANAGEMENT

Dr. William A. Hornfeck
Gannon College
Erie, PA

Mr. John Cavin
Mr. Ron Boulet
Air Force Armament Laboratory
Eglin AFB, FL

## ABSTRACT

Current approaches to aircraft design have identified stores management as a distant function entity. In general, aircraft stores are defined as the complement of weapons and suspension devices which are intended as line replaceable. The management of aircraft stores involves the control and communications system necessary in providing the pilot efficient interaction with the aircraft stores which are present.

Recent technology advances have provided the impetus to optimize the design and integration of Stores Management Systems (SMS) with two principal goals in mind: (1) reducing the complexity of the hardware required without sacrificing the capability and reliability of stores management features; and (2) increasing the range of applications of stores management systems in order to reduce the number of custom aircraft/stores configurations.

This paper discusses alternative stores management configurations and proposes a modular data system architecture utilizing a nested arrangement of MIL-STD-1553B multiplex data bus links. The rationale for the selection is based on the opportunity for future SMS configurations to capitalize on current and anticipated breakthroughs in related technologies.



PRECEDING PAGE BLANK

## I. INTRODUCTION

This paper will document the results of a research effort sponsored by the Air Force Office of Scientific Research and performed in support of the Armament Laboratory at Eglin Air Force Base. The research has specifically involved problems related to aircraft Stores Management Systems (SMS) and associated technologies.

- centralization of stores management functions,
- functional partitioning of stores management functions, and
- transparency of pilot-to-store and store-to-pilot interface;

as well as system concepts such as:

- enhancement of reliability and confidence levels,
- aircraft/stores interoperability, and
- minimization of pilot recognition/reaction time.

Perhaps the pre-eminent issue involving aircraft/stores interface is one of rising life-cycle costs associated with growing set of aircraft and stores having limited interoperability characteristics. This situation has evolved primarily because interface standards have been nonexistent and stores have traditionally been designed for specific aircraft/ applications.

A solution to the interoperability problem will involve a well-conceived set of standards and some radical departures from established design strategies for Stores Management Systems. In addition, a careful assessment of technology trends must be undertaken to pinpoint those technologies which should be further encouraged by the Air Force, and where technology breakthroughs should be anticipated. An SMS to address the needs of aircraft for the 1980's, 1990's, and beyond will need to be flexible as well as efficient (goals which are usually incompatible) and must take advantage of the most current systems and hardware technology.

## II. BACKGROUND AND RATIONALE

An aircraft Stores Management System must provide a suitable interface between the pilot and the full complement of aircraft stores and associated suspension equipment which is onboard. This interface must be capable of providing the pilot with the communications, control, and display functions necessary for efficient management of onboard stores during

all phases of a particular mission. Figure 1 illustrates the conceptual problem associated with providing this interface.

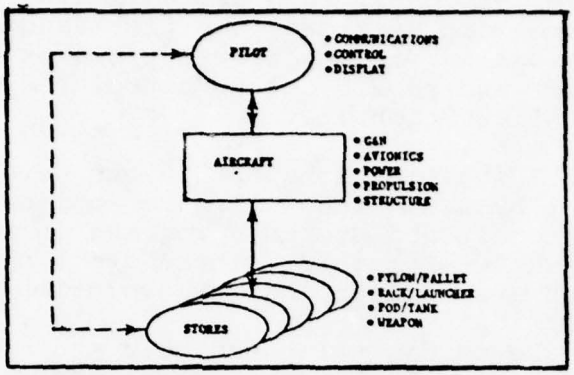



Figure 1. Virtual and Actual Pilot-Stores Interface.

The solid connecting lines indicate the hardware interfaces which are present and necessary for the integration of pilot/aircraft/stores functions and operations. The dashed connector shows a "virtual" interface which reflects direct pilot interaction with onboard stores. An efficient Stores Management System will provide the appearance of virtually direct control but will utilize all appropriate aircraft subsystems and interfaces in performing the Stores Management functions.

If one begins with the attributes which are characteristics of an over-all Stores Management System, the following list includes the principal items which must be addressed:

- effective centralization of the functions associated with stores management;
- continuous pilot appraisal of the complete inventory of stores onboard;
- a flexible but efficient means for control of any of the available stores' operational modes;
- maximum interoperability of available aircraft and stores;
- transparency of pilot-to-store and store-to-pilot interface;
- minimum pilot recognition/reaction time;
- enhancement of system reliability and confidence levels.

To a fairly high degree, the attributes listed above are achievable with the exception of the fourth item--interoperability of aircraft and stores. Unfortunately, it is the interoperability attribute which has the most drastic effect on the operational costs of an aircraft/stores suite. These op-

erational costs can be considered to include penalties which are not strictly dollar costs, such as: a limited set of aircraft on which stores can be flown; the time associated with flight-time operations in fitting aircraft/stores configurations; and logistics and repair delays associated with custom aircraft/stores configurations.

Despite the difficulties inherent in the development of a Stores Management System which will have application across a wide range of aircraft and stores, progress in related areas in making this objective feasible. There are two specific design trends which are influencing the evolution of Stores Management Systems:

(1) Increased use of digital techniques and microelectronics devices. This trend places added emphasis on distribution of signal and data processing, digital multiplex data bus communication, and highly modular hardware systems.

(2) The use of increasingly information-intensive system. This results in the realization of a greater proportion of functions via software and closer attention to the application of engineering principles to all aspects of the software life cycle.

There are a number of current digital multiplex communications/processing systems in various stages of development which reflect the trends outlined above. A representative list includes:

- F-15 Avionics Data Bus — operational
- F-16 Stores Management Set
- F-18 Redundant Multiplex Data Bus
- B-1 Data Multiplex Buses
- Space Shuttle Multiplex Data Bus System
- Navy Shipboard Data Multiplex System (SDMS)

— Developmental

- Eglin F-4E SMS Flight Test Program
- AFAL DAIS Multiplex System
- Integrated Digital Avionics for Medium STOL Aircraft (IDAMST)
- Advanced Remotely Piloted Vehicle (ARPV)

— Investigative

This list represents a group of bellwether systems from which valuable lessons can be learned regarding systems design for advanced military aircraft. By no means is this group inclusive of all systems which offer data points useful in determining directions for evolving Stores Management Systems. Other sources include industrial systems, telecommunications systems, programming systems, and so on. The critical consideration

which should be given to advanced SMS development is to avoid a reliance on serendipity for the evolution of improved systems. Rather, careful attention should be given to the areas of research and development which should be encouraged for future applications.

## III. SYSTEM STUDIES

### SMS ARCHITECTURE

Data communications/processing architectures have undergone considerable changes during recent years. These changes have resulted from the increased capability of hardware and better understanding of communications techniques. For many years, the advantages of distributed architectures were apparent; however, the realization of effective distributed systems had to await the arrival of modular computer systems, microprocessor technology, and a better understanding of computer networking strategies. This now being the case, not only effective distributed systems, but also federated or embedded architectures are somewhat commonplace.

The system designer;s problem has changed from one of adapting a solution to an available architecture to one of selecting the appropriate architecture for a problem solution. Not only are the classical architectural concepts being used, "hybrid" architectures using aspects of more than a single type are being implemented, as well as "custom" architectures devised for a particular application.

This freedom of architectural form allows the development of a Stores Management System concept which is tailored to functions and constraints of the aircraft/stores environment. A principal consideration is the MIL-STD-1553B multiplex data bus. This standard is an effective compromise between the sophistication required for efficient communications links and the simplicity required for modularization and interoperability. In this paper, it will be assumed that a MIL-STD-1553B multiplex digital data bus will be used for all communications; however, the topology of the computer/communications network will be tailored to the SMS function.

A second set of assumptions which will influence an SMS architecture involves the physical characteristics of the aircraft/stores configurations to be considered. Figure 2 shows the three basic configurations assumed as representative of virtually all aircraft/stores which are of interest.

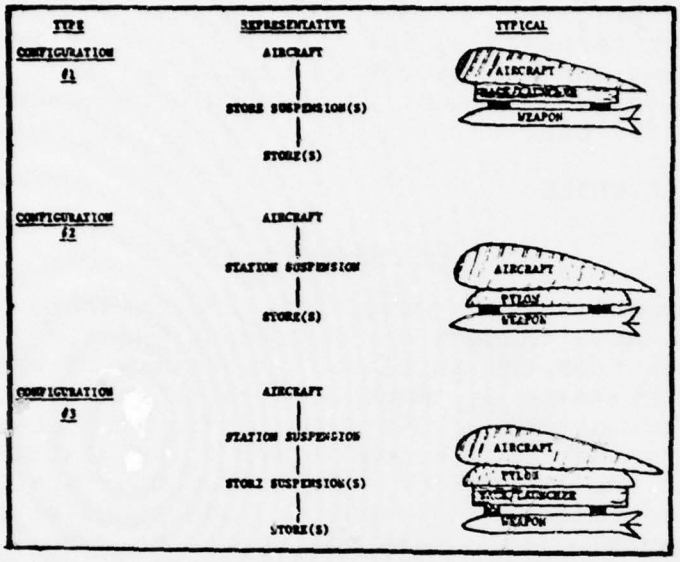



Figure 2. Basic Aircraft/Store Configurations.

A final set of assumptions considers the communications paths which are required to accomplish the Stores Management function. First, it will be assumed that avionics data will be required by the SMS and therefore data transfer (two-way) between the avionics bus and SMS bus must be possible. (This also assumes a bus structure for the avionics subsystem.) Secondly, all independent buses will require a bus controller. Third, all digital data interfaces are MIL-STD-1553B compatible. All units connected to a single bus must appear as remote terminals and any SMS remote terminal may communicate with any other SMS remote terminal.

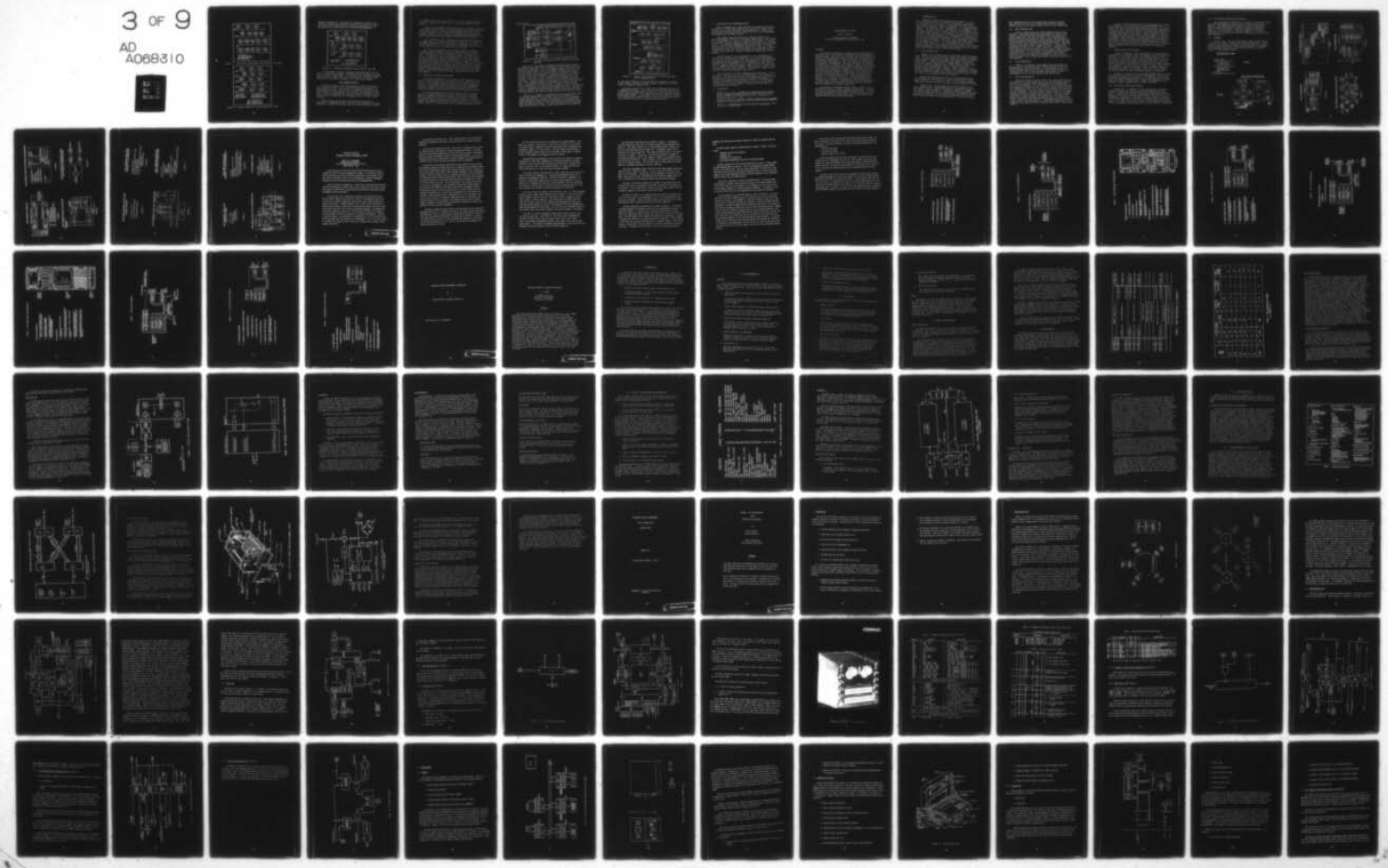
The simplest architectural concept for SMS consideration would be a single SMS multiplex bus with all related subsystems appearing as remote terminals. This structure is shown in Figure 3 and can be viewed as a "horizontal" bus structure.

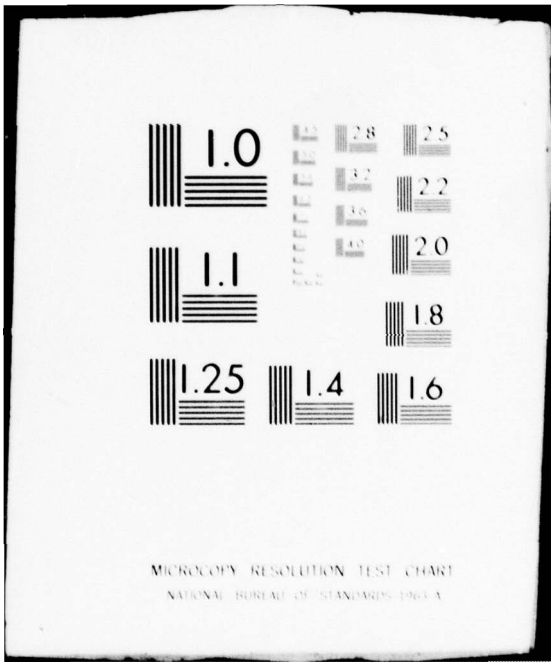
An SMS architecture which could be considered as a "vertical" bus structure is shown in Figure 4. This architecture allows independent communication on three separate, but interconnected, SMS buses plus an interface to the aircraft avionics bus and any weapon buses.

In addition to the strictly horizontal or vertical structures, there are a large number of "diagonal" bus structure possibilities. In general, such a diagonal structure would increase communications speed at the expense of hardware and

AD-A068 310 AERONAUTICAL SYSTEMS DIV WRIGHT-PATTERSON AFB OH F/G 17/2
PROCEEDINGS OF THE AFSC MULTIPLEX DATA BUS CONFERENCE (2ND), HE--E
OCT 78 E C GANGL
UNCLASSIFIED ASD-TR-78-34-VOL-2 NL

3 OF 9
AD A068310




1.0
1.1
1.25
1.4
1.6
2.8
2.5
2.2
2.0
1.8
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

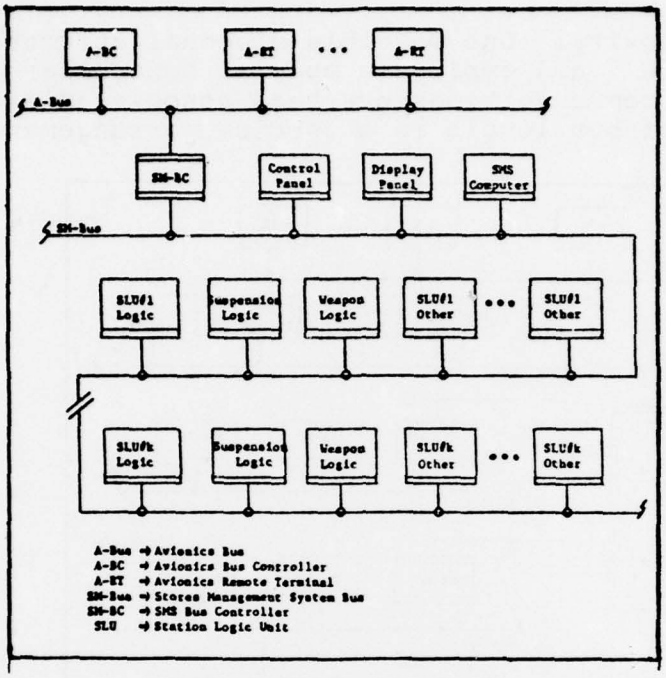



Figure 3. Universal (Horizontal) Bus Structure for SMS.

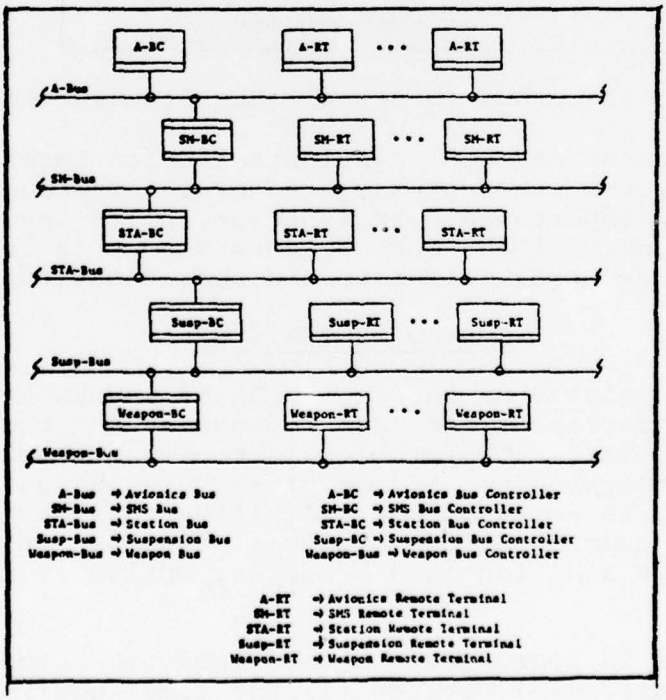



Figure 4. Nested (Vertical) Bus Structure for SMS.

software complexity. One possible diagonal structure is shown in Figure 5 and employs a special controller in order to reduce the communications overhead associated with the multiplicity of bus levels in a vertical arrangement.

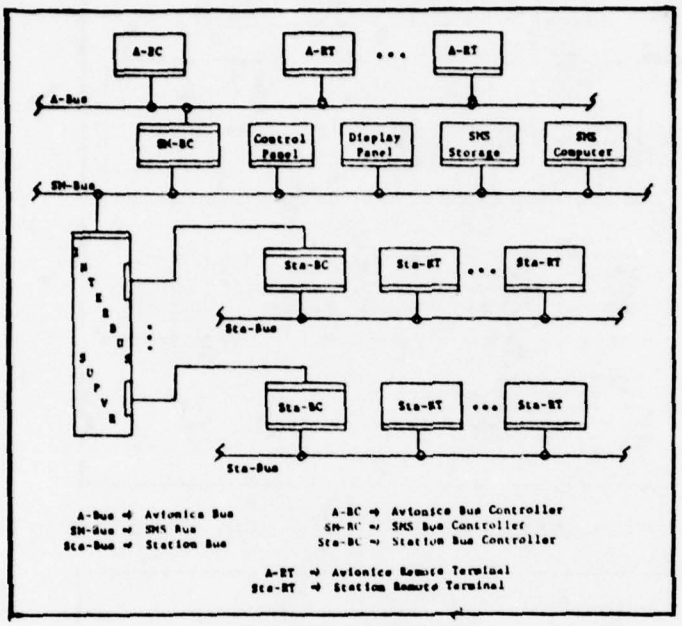



Figure 5. Interbus (Diagonal) Bus Structure for SMS.

An assessment of these candidate architectures must involve not only the inherent advantages/disadvantages, but also their relative importance. In addition, all relevant technical aspects of an overall SMS must be considered, including communications, hardware, software, and operations support.

## SMS COMMUNICATIONS

Directly related to the selection of an SMS architecture is the specification of the data communications procedures which shall be used. The problem is somewhat bounded by the assumption concerning use of the MIL-STD-1553B multiplex data bus. However, this standard still allows considerable flexibility in the communications protocol which is employed, especially with the addition of a broadcast option in the 1553B version.

Choice of an appropriate communications protocol is generally a compromise between (1) the time required for transmission of a complete message from one communications terminal

to another and (2) the complexity of a network which minimizes message transmission times for all possible communications paths.

The actual communications protocol to be followed imposes rules upon central, area, or local communication nodes with the objectives being an orderly and reasonably efficient flow of network messages. If a time-division-multiplex data bus is assumed, three principal types of communication protocol would merit investigation: Round-Robin, Polling, and Interrupt.

It is possible, for a particular communications network, to adopt one of the above protocols exclusively. In addition, a communications structure may employ a combination of the three possibilities. The selection is dependent upon the characteristics desired in a particular application.

For Stores Management System applications, a command-response polling protocol would be compatible with MIL-STD-1553B and would provide a reasonably simple control structure. In addition, testing and validation are simplified due to the synchronous nature of communication. Message transmission speeds are critical constraints. On the other hand, an interrup protocol would provide a speed advantage but an additional interrupt bus would be required. Also, an interrupt system is asynchronous in nature and testing/validation procedures would require considerable effort. Finally, it should be mentioned here that a round-robin protocol provides advantages of synchronous operation but this protocol is not well suited to the configuration changes associated with the SMS environment.

## IV. SELECTION OF SMS ARCHITECTURE

Effective development of an adaptive SMS would necessarily be in the context of anticipated aircraft technology evolution. The diagram shown in Figure 6 is a graphic description of the technical trends affecting the Stores Management function. The blocks which are shaded represent technology driving functions and the overall diagram illustrates the evolution toward integrated aircraft/avionics/weapons systems.

A Stores Management System architecture which is best suited to highly integrated aircraft design with a maximum degree of modularity would be based on the nested architecture described in Section III and shown in Figure 4. Figure 7 shows the nested architecture in greater detail. From this figure, correspondence with the physical aspects of an aircraft/stores interface is observable and the inherent modularity can

also be seen.

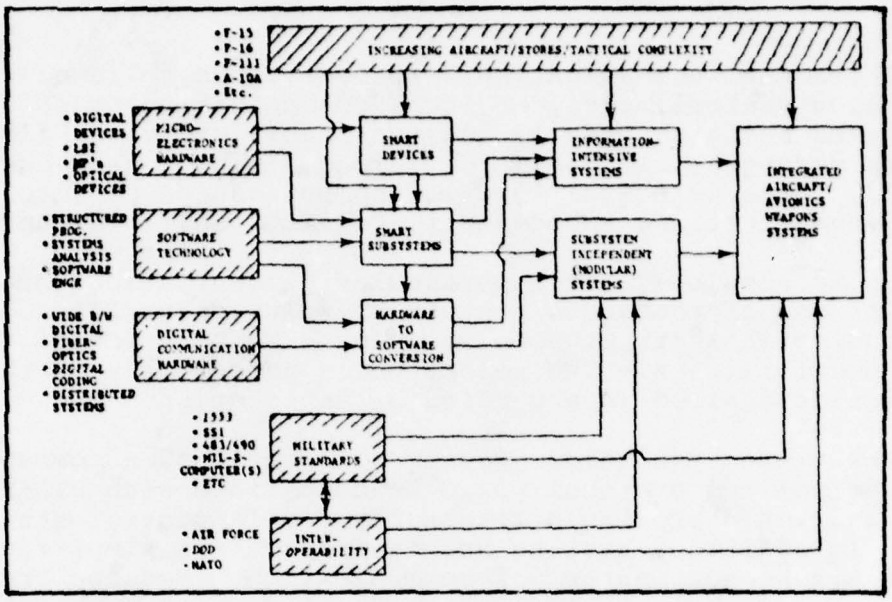



Figure 6. Aircraft Technology Trends Toward Integrated Systems.

The core of the Stores Management System would include the SM-BUS with associated terminals and the STA-BUS with its associated terminals. In a development sense, the store suspension architecture as well as weapon architecture would be the responsib lity of the appropriate development programs. Central to the interoperability features would be a standard multiplex data bus interface, communications protocol, and software structure. These concepts are presently being employed at the avionics bus level and the proposed architecture woul d require similar communications in a nested arrangement. The communications rate problem is encountered when data transfers are required across a number of bus levels. In this situation, the efficiency provided by the bus controller units in transferring information becomes critical. Bus controller capacity is also the issue when considering the overhead associated with data transfer between and two bus levels.

There are a number of technological and operational considerations which tend to support the investigation of a nested architecture. First, the nature of stores management communications is such that the necessity for transferring large volumes of data at high rates across several bus levels does not presently exist. In the event that future requirements dictate such a need, a low-risk posture is afforded by

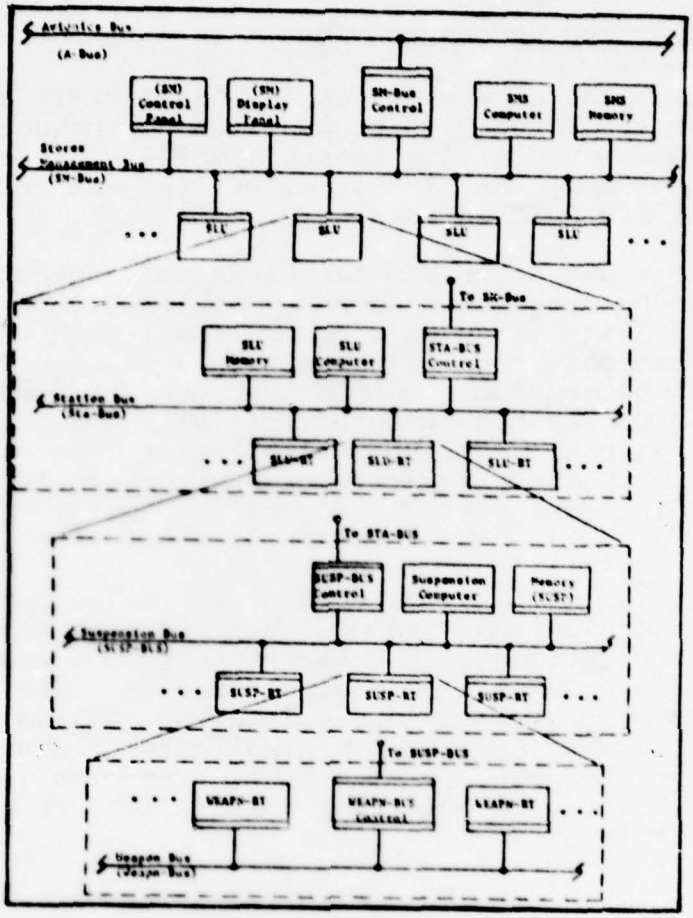



Figure 7. Nested Bus Architecture for Avionics/SMS/ Stores Communication.

the potential benefits of technological advances in fiber-optics links, component speeds, memory capacities, and processor throughput capabilities.

Efficient control of overall SMS communications will be a formidable problem. This control will reside principally in SM-BUS level components but will be distributed to a large extent among station level components as well as lower-level processing systems. Ultimately, the effectiveness of the control schemes will depend in large part on the software structure adopted during the conceptual design stages.

## V. CONCLUSIONS AND RECOMMENDATIONS

The development of a wide variety of aircraft and stores to fulfill the Air Force mission has created the need for solutions to problems involving interoperability. These problems concern not only the Air Force, but the entire Department of Defense and the NATO nations.

This report has described the technical aspects of a Stores Management System for advanced applications which treats the interoperability issue as a design focal point. In doing so, several critical technical aspects are exposed for consideration. The computer/communications network which would be best suited for the Stores Management function is one. A nested architecture has been described which has an inherent physical compatibility with the aircraft/stores structure and is considered to be the most suitable scheme for immediate investigation.

The selection of an architecture is not independent of the optimum communications protocol for a Stores Management System. It is pointed out that both synchronous and asynchronous communications schemes bear consideration. The speed advantages of an asynchronous (e.g., interrupt-driven) system must be compared with the greater simplicity of a synchronous (e.g., command-response) system. Other tradeoffs include consideration of additional hardware, testability, compatibility with 1553 multiplex standard, and so on.

The selection of an SMS communications protocol, in turn, will have a great effect on the complexity of the required software. The adaptive SMS which is suggested in the previous section would rely heavily on the effective management of software development and mainenance efforts. A sufficiently complex software structure would pose a threat to the basic feasibility of such an attempt to enhance aircraft/stores interoperability features.

## VI. REFERENCES


1. Barnes, B., et. al., Analysis of Hardwired Versus Multiplexed Stores Management Systems, AFATL Technical Report 76-73, Volumes I, II, Harris Electronic Systems Div., Melbourne, FL, July 1976.
2. Military Standard, Aircraft Internal Time Division Command/Response Multiplex Data Bus, draft MIL-STD-1553B, June 1978.
3. Mennie, D., Communications and Microwave Technology, IEEE Spectrum, January 1978.

# MULTIPLEXING IN AIDS

John F. Moran

General Electric Company
Aircraft Equipment Division, Utica, NY

ABSTRACT

This paper describes the application of multiplexing techniques within the Navy's Advanced Integrated Display System (AIDS). The AIDS project under the direction of the Naval Air Development Center is currently in the Advanced Development Model phase. General Electric is system contractor with Intermetrics, Inc. the associate contractor for software development. The goal of AIDS is to develop a modular set of displays and controls along with associated processing and data multiplexing elements that can be applied to all future Navy aircraft. Objectives are improved pilot performance, reduced weight and Life Cycle Cost and improved mission reliability. The extensive utilization of multiplexing is a key factor in helping to achieve all of these objectives. Multiplexing will be applied in three distinct areas: (1) narrowband serial digital (i.e., 1553B), (2) wideband video suitable for transmitting TV Raster formatted data, and (3) high-speed parallel digital suitable for transmitting data between digital modules on a common back panel.

The Advance Development Model (ADM) of AIDS will be implemented with currently available technology. It will be used to validate system concepts and software. It will also be designed with sufficient flexibility so that advanced multiplexing technologies can be incorporated and demonstrated as they become available.

## I. INTRODUCTION

Advanced Integrated Display System (AIDS) is a U.S. Navy project currently in Advanced Development (Figure 1). It is directed by the Naval Air Development Center (NADC) with General Electric Company responsible for system development and Intermetrics, Inc., responsible for software development. Its objectives are improvements in performance, reliability, weight and life cycle costs in the display/controls of future Navy aircraft. This paper first describes briefly the basic AIDS configuration and program plan as well as the objectives of AIDS and how their achievement is dependent on the use of multiplexing techniques. The subsequent portions of the paper describe in turn the three areas where multiplexing is applied in AIDS; (1) Narrow Band Serial Digital, (2) Wide Band Video, and (3) High Speed Parallel Digital. Each of these sections addresses the requirements, the planned architecture, the initial ADM implementation and where appropriate some growth options that are available.

## II. AIDS DESCRIPTION

Figure 2 is a cockpit view of the basic AIDS configuration. The pilots basic displays include the Heads-Up Display (HUD), Vertical Situation Display (VSD), Horizontal Situation Status Display (HSD) and two identical Status Advisory Displays (SAD). A Helmet Mounted Display/Sight is optional. Principal control elements are the Mode Selector Panel (MCP) and the Integrated Control Panel (ICP) which provides switches with programmable legends. These legends are displayed using Liquid Crystal Display (LCD) technology. Another cockpit element not shown on Figure 2 is the Briefing Information Entry Device (BIED) which allows preflight entry of mission data via a magnetic cartridge device.

An optional sensor station is also shown which would, for example, be applicable to missions envisioned for the Navy's V/STOL A. The objective here is to maximize hardware commonality between pilot and sensor station displays/controls.

Figure 3 is a highly simplified block diagram of the total AIDS system. All display/control electronics outside the cockpit will be housed in Modular Integrated Display Electronics Racks (MIDER). Connection between the MIDERS and the cockpit elements of AIDS and between the MIDERS and other Avionic systems will be via multiplex systems both narrow band digital (1553B) and wide band video. It will be noted

that communication to the imaging type cockpit displays involves both 1553B and the wide band multiplex systems. Communication to all other cockpit units is via 1553B only.

## III. AIDS PROGRAM PLAN

Figure 4 summarizes the AIDS background and project schedule. Numerous exploratory development projects over several years provided the background for the AIDS system design which was initiated in 1976. This system design identified the most promising technologies, established the baseline system configuration described herein and produced specifications for the Advanced Development Model. This ADM is now in the early phase of implementation. Human factors considerations are a vital aspect of AIDS. The ADM when completed in 1981 will be used at NADC for integration with other advanced avionics systems and to support human engineering experiments. It will also be designed with sufficient flexibility to later incorporate advanced technologies. These will likely include such advances as color and flat panel displays, solid state mass memory and fiber optics data transmission.

## IV. AIDS OBJECTIVES

Figure 5 illustrates the primary AIDS objectives as well as their relationship with several concepts that are fundamental to the AIDS concept. Multiplexing is one of these concepts and it will be seen that either directly or indirectly the use of multiplexing contributes to the achievement of all of the primary AIDS objectives.

## V. AIDS CONFIGURATION

Figure 6 shows in greater detail the AIDS baseline configuration and illustrates some of the architectural features that have an impact on the requirements for multiplexing. One of these features is the distribution of processing using microcomputers, which has the effect of reducing buss traffic. For example each unit will control its internal self test via its microcomputer and report only summary status to the AIDS control processor. Another feature is the location of symbol generators at the displays where feasible. The only symbol generators located outside the displays will be those for complex in-raster symbology associated with the imaging displays. These will be located in the MIDER and their output is transmitted to the displays via a wide band video buss. Consequently the traffic on the internal AIDS 1553B is relatively light.

Figure 7 depicts AIDS from a larger perspective showing its interfaces with other Navy Avionic subsystems many of which are also in the Advanced Development phase. It will be noted that the interfaces consist of the wide band multiplex carrying sensor video and the main avionics buss (1553B) for transmitting serial digital data. The postulated avionics configuration does not include a central computer; rather it is assumed that processing is distributed among the various subsystems. AIDS therefore includes sufficient processing and memory capacity to perform all display/control functions. The result is that the data rate between AIDS and the main avionics buss is low relative to what it would be if display/controls computations were being performed in a Central Computer.

## VI. NARROW BAND MULTIPLEXING

Figure 8 summarizes the requirements for narrow band multiplexing both within AIDS and between AIDS and other subsystems. It is expected that the broadcast mode will be utilized for functions such as distributing data of interest to several users, distributing mission time, etc. Figure 9 shows the planned architecture for these 1553B busses. Redundant busses are provided to insure that no single failures will result in a total failure of AIDS. Cockpit units are interfaced to both 1553B busses: each unit will, under non-degraded conditions, be assigned to one of the two busses. The second buss is used for degraded modes of operation.

Figure 10 is the initial terminal implementation in the AIDS ADM using currently available components. In so far as possible the design will factor in the configuration of the BIU chips under development such that later versions of AIDS might incorporate these chips. Figure 11 summarizes other possible growth considerations in the 1553B area.

## VII. WIDE BAND VIDEO MULTIPLEXING

Figure 12 is a summary of the requirements including projected growth for the wide band video multiplex buss. Figure 13 shows the planned architecture where again redundancy is provided to prevent catastrophic AIDS failures. The system will initially be implemented with readily available commercial components to provide a low cost vehicle available at an early date that can be used to demonstrate system concepts. At a later date this system can be updated to incorporate more advanced technologies such as fiber optics (Figure 15).

## VIII. HIGH SPEED PARALLEL MULTIPLEXING

The requirements summarized in Figure 16 represent those for the AIDS Modular Integrated Display Electronics Rack (MIDER). The implementation shown in Figure 17 provides a much greater data transfer capability than is required by AIDS ( 200K words/sec). This high speed multiplexing approach is not unique to a display system: it could be applicable to other digital avionics employing an integrated rack approach.

## IX. SUMMARY

(Figure 18). The AIDS ADM makes extensive use of multiplexing in order to achieve the system objectives. While in some areas readily available technology will be initially used, the designs will be sufficiently flexible to accommodate advanced technologies such as fiber optics as they achieve a more mature status.

**MULTIPLEXING IN AIDS***

- AIDS SUMMARY
- MULTIPLEXING DRIVERS IN AIDS
  - SYSTEM OBJECTIVES
  - SYSTEM ARCHITECTURE
- AIDS MULTIPLEXING DESCRIPTION
  - NARROW BAND SERIAL
  - WIDE BAND VIDEO
  - HIGH SPEED PARALLEL
- SUMMARY

* ADVANCED INTEGRATED DISPLAY SYSTEM

FIGURE 1

**AIDS COCKPIT CONFIGURATION**

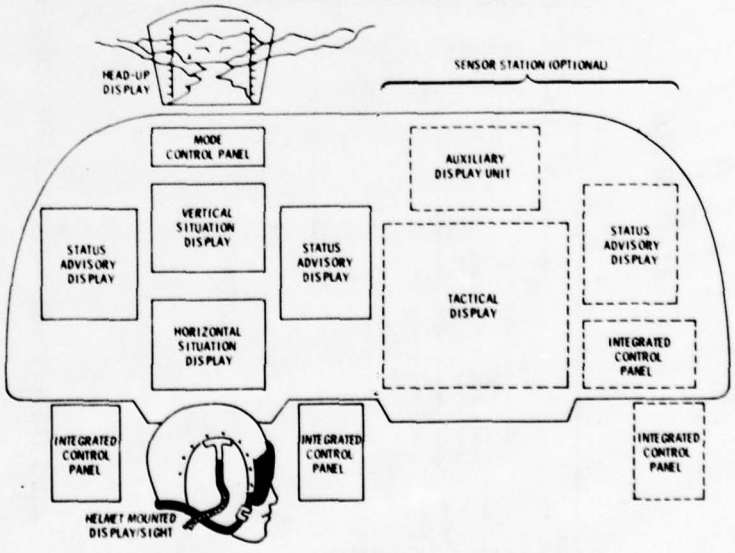



FIGURE 2

## AIDS ARCHITECTURE

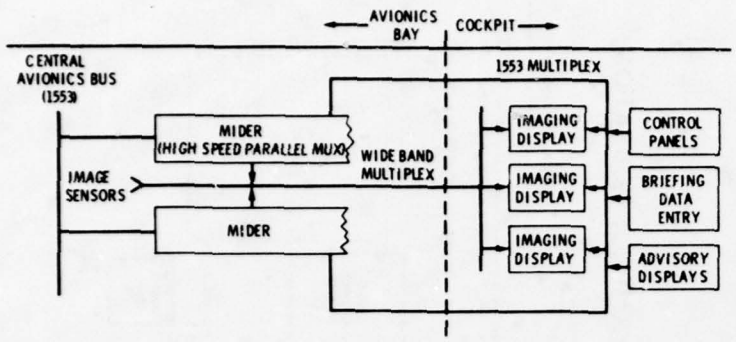



FIGURE 3

## AIDS PROGRAM SCHEDULE

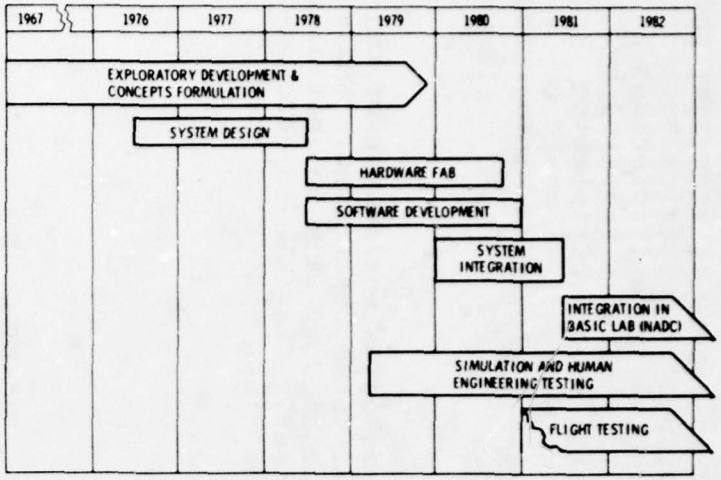



FIGURE 4



## AIDS OBJECTIVES AND RELATED CONCEPTS

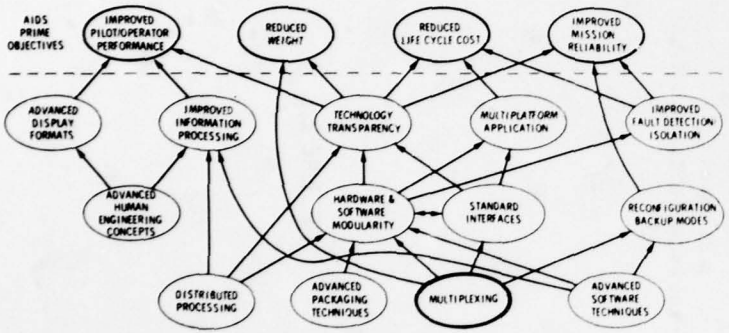



FIGURE 5

## AIDS CONFIGURATION

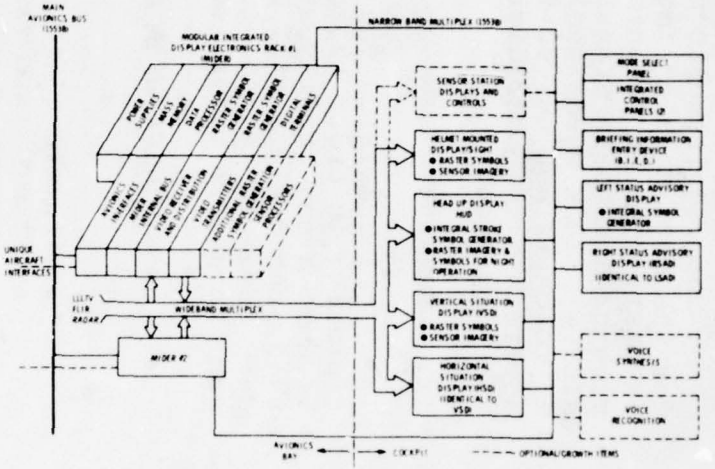



FIGURE 6

## AIDS EXTERNAL INTERFACES

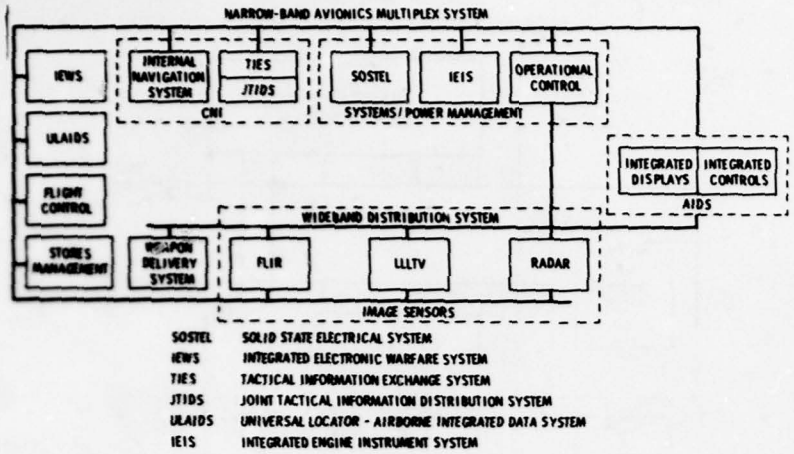



FIGURE 7

## AIDS NARROW-BAND MULTIPLEXING REQUIREMENTS

| | INTERNAL | EXTERNAL |
|---|---|---|
| DATA RATE (WORDS/SEC) | 5-15,000 | 2,000-20,000 |
| COMPATIBILITY | 1553B | 1553B |
| BROADCAST MODE | YES | YES |
| NUMBER OF TERMINALS | 8-20 | 2-4 |
| DYNAMIC BUS ALLOCATION | YES | NO |
| HARDWARE REQUIREMENTS | • STANDARD TERMINAL MODULE<br>• INTERFACE WITH AIDS MICRO-COMPUTER | |
| REDUNDANCY | SYSTEM OPERATIONAL AFTER ANY SINGLE FAILURE | |
| BITE | DETECT FAILURES AND ISOLATE TO MODULE | |

FIGURE 8



## AIDS NARROW-BAND MULTIPLEXING

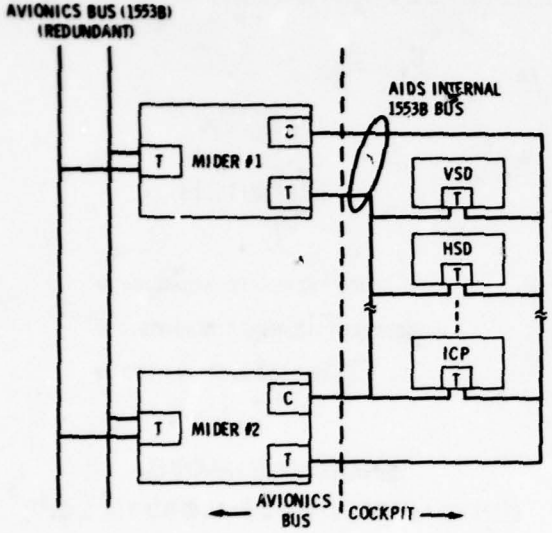



FIGURE 9

## AIDS ADVANCED DEVELOPMENT MODEL 1553B TERMINAL IMPLEMENTATION

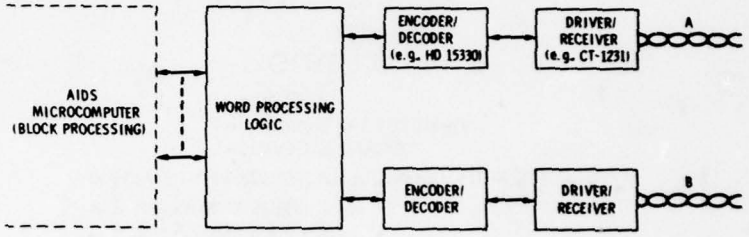



FIGURE 10

## AIDS NARROW-BAND MULTIPLEXING GROWTH OPTIONS

- UTILIZATION OF BIU CHIPS
- FIBER OPTICS TRANSMISSION MEDIA
- PACKAGING ON ISEM 2A MODULE

FIGURE 11

## AIDS WIDEBAND VIDEO MUX SUMMARY OF REQUIREMENTS

- MAXIMUM OF 16 CHANNELS
- TRANSFER DATA IN RS-170/EIA 343 RASTER FORMATS
- 17 MHz CHANNEL BANDWIDTH
- 25 MHz CHANNEL SEPARATION
- SUPPORT AIDS REDUNDANCY/RECONFIGURATION CONCEPTS
  - TUNABLE RECEIVERS
  - ALTERNATE BUS SELECTABLE

FIGURE 12

## AIDS WIDEBAND VIDEO MUX ARCHITECTURE

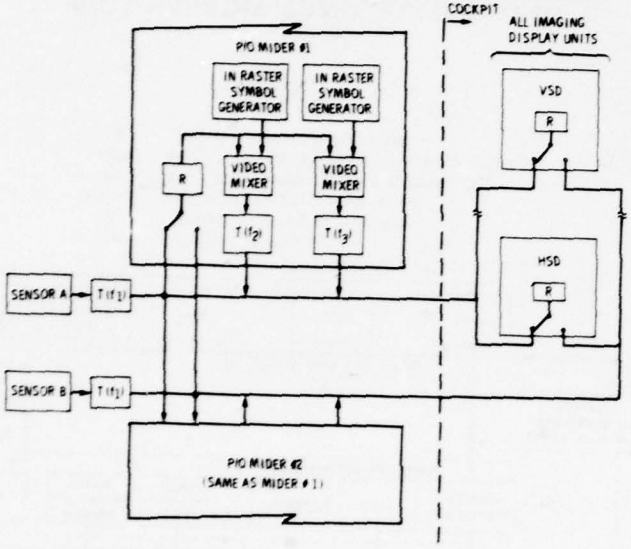



FIGURE 13

## AIDS WIDEBAND VIDEO MUX INITIAL ADM IMPLEMENTATION

- RF MULTIPLEXING
- OFF-THE-SHELF COMPONENTS
  - TRANSMITTERS
  - RECEIVERS
  - POWER SPLITTERS
  - SUMMING ELEMENTS
- RECEIVERS MODIFIED FOR DIGITAL TUNING
- INITIAL OBJECTIVES
  - VALIDATE SYSTEM CONCEPTS
  - CHECKOUT OF SOFTWARE
  - VEHICLE FOR DEMONSTRATING ADVANCED TECHNOLOGIES

FIGURE 14

## AIDS WIDEBAND VIDEO MUX GROWTH OPTIONS

- WIDER RF CHANNELS
- FIBEROPTICS MEDIA
  - RF MULTIPLEXING
  - COLOR MULTIPLEXING

FIGURE 15

## HIGH-SPEED PARALLEL MUX SUMMARY OF AIDS REQUIREMENTS

- 8-14 USERS
- 30,000 WORDS/SEC PEAK DATA RATE
- 16 BITS/WORD
- USER-USER AND USER-CONTROLLER MODES
- COMPATIBLE WITH INTEGRATED RACK CONCEPTS
  - USERS ON COMMON BACKPANEL
  - ADDITION OF USERS WITH MINIMUM IMPACT

FIGURE 16

## HIGH-SPEED PARALLEL MUX IMPLEMENTATION

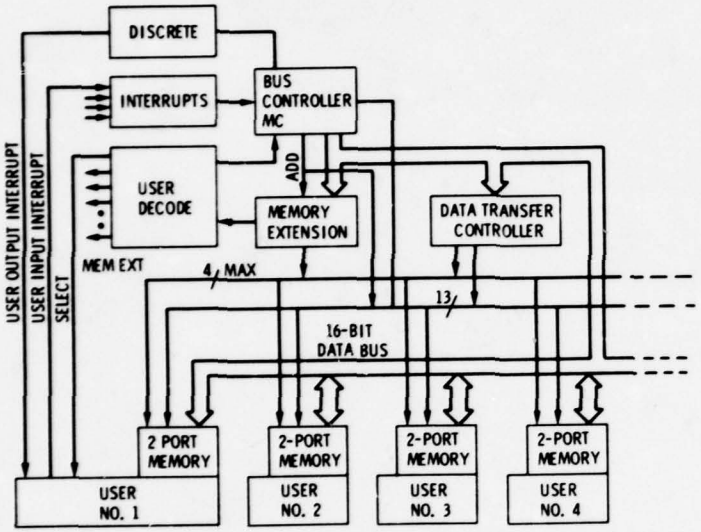



FIGURE 17

## MULTIPLEXING IN AIDS*

- AIDS SUMMARY
- MULTIPLEXING DRIVERS IN AIDS
  - SYSTEM OBJECTIVES
  - SYSTEM ARCHITECTURE
- AIDS MULTIPLEXING DESCRIPTION
  - NARROW BAND SERIAL
  - WIDE BAND VIDEO
  - HIGH SPEED PARALLEL
- SUMMARY

* ADVANCED INTEGRATED DISPLAY SYSTEM

FIGURE 18

# MULTIPLEX BUS FOR INTEGRATED AVIONICS MANAGEMENT SYSTEMS

Merrill T. Ludvigson
Rockwell-International
Collins Government Avionics Division
Cedar Rapids, Iowa

In my discussion of the use of the MIL-STD-1553 multiplex bus as an integration tool for Avionics Management systems, I will begin by taking a look at multiplexing from an historical perspective, then proceed to the main body of the paper to the use of multiplexing in Avionics Management Systems and follow with some conclusions and recommendations.

From an historical perspective, I would like to review briefly the motivation for multiplexing, discuss some of the constraints which have limited the breadth of application of multiplexing and review the experience gained by Rockwell-Collins and others.

I am sure that we all recognize the important role multiplexing can play to reduce complexity and weight of aircraft interconnect and the role it can play to improve integration opportunities with growing digital processor applications. Another important role is that of the integration of controls and displays to simplify the relationships between crew station control/display functions and aircraft systems being managed. But perhaps one of the most important roles of all is to provide the flexibility needed to allow the weapon system to be upgraded for changing requirements. Funding is getting tighter, yet technology is progressing at an ever increasing rate while the acquisition time appears to be lengthening. All of this only serves to increase the probability that threats will change before a weapon system becomes operational.

PRECEDING PAGE BLANK

But certain constraints exist. One is the availability of MIL-STD-1553 compatible equipment and the other is the availability of suitable industry standards.

But why do the constraints exist? First of all, previous weapon systems utilized multiplex technology for limited subsystem application. Airframes developed centralized systems and utilized similar but unique bus techniques, and we as vendors responded to specific program specifications. As a result, integration required a wide variety of special adapters and concentrators to interface widely variant sensors and standard avionics equipment.

But obviously considerable experience has been gained from the systems and equipment numerous airframes and avionics suppliers have produced to date. The Collins background has been aided from the vantage point of a supplier of multi-user product lines which include Comm/Nav equipment, cockpit management systems, flight instruments; and as a subsystem contractor for such programs as Integrated Avionics Control System (IACS), the F-15 digital HSI (Horizontal Situation Indicator), on the space shuttle digital HSI program and a number of aircraft flight directors. Collins background also includes experience as an avionics integrator in such programs as the HU-25A, Medium Range Surveillance Aircraft (MRS) and the USCG HC-130H Cockpit Management System. Playing these multiple roles has given Collins a varied insight into such important considerations as system design management for both hardware and software, system partitioning for integratability and for functional integrity, bus utilization for bandwidth conservation and for redundancy, the impact multiplexing has on product line standards and the cost of ground support equipment, and the detailed specifications required to assure integration of various suppliers equipment.

To properly address the application of the multiplex bus as an Avionics Management System, we will address such factors as system content, operational considerations, that is, those special considerations necessary to see that that the system performs the desired functions when everything is working; and system integrity, that is, those considerations necessary to see that it can perform when something fails. We will then review four examples with these considerations in mind.

The first thing one must do is to determine the system contents. What makes up a shipset? There may be a diversity of functions such as control and display and/or integration of entire subsystems, such as communications, navigation, weapons, or flight control. One must identify such factors as multiple operator station capability and be aware of the degree of compatibility of the elements to be managed to the bus concept.

The operational considerations have to do with such factors as operator manipulation or the times, responses, and priorities related to the human interface, as well as such purely hardware considerations as system performance via the bus, that is, bus loading, delays, or timing, and what part partitioning may play in resolving these and other problems.

The system integrity considerations include identifying the survivability needs of the various components or subsystems and determining what role is played by the management subsystem. How can partitioning, redundancy, and the use of reversionary modes be used to ensure system integrity? Bus protection is an important consideration which can benefit from such techniques as hardware or software time out and by physically protecting the bus by the use of external bus couplers and physical separation of the buses.

The first of the four systems I will describe is the U. S. Army Integrated Avionics Control System (IACS) program shown in figure 1. You will note the special emphasis given to the bus so that the similarity between this and other systems is readily apparent. The system design is based on MIL-STD-1553A and provides a dual bus for redundancy and redundant bus controllers. Interface adapter modules are used for non-compatible avionics equipment and backup system computers are provided. You will also note interface capability with doppler radar, the mission computers, and other equipment, if necessary.

The IACS is a cockpit management system which currently performs control display functions. (See figure 2) While the current implementation includes a primary control unit, secondary control unit, and a status panel, the system can accommodate another primary control unit and two more status panels. The shipset of equipment being managed is shown and none of the equipment have been designed to be MIL-STD-1553 compatible.

The operational considerations are shown in figure 3. The operator interface designs are the result of extensive human factors background and analysis. The primary control unit is a multipurpose control display unit which utilizes multipage display capability. A secondary control unit is provided with reduced capability. More than one primary control unit may be used with the system, if necessary. The IACS configuration is capable of dedicated bus operation and is also capable of dynamic bus allocation using an external bus controller. The bus loading is less than 10% with a service rate of 10 times per second. The critical timing is associated with system reaction to operator keystroke or function selection.

The Cockpit Management System (CMS-80) for the Medium Range Surveillance (MRS) aircraft is shown in figure 4. You will note the similarity of the basic bus architecture. You may note also the addition of several new equipments, such as the HF, DME, and TACAN and the navigation computer which, in turn, interfaces with the navigation sensors and the mission package.

The basic architectural approach for the MRS is also based on MIL-STD-1553A with dual bus for redundancy and dynamic allocation for efficient bus tuilization. In this application, the bus is used for map transfer from the navigation computer to the Control Display Unit (CDU).

As shown in figure 5, the system performs the control and display functions for the flight and fuel management as well as the communications and navigation equipment. Interface adapter modules are necessary for all equipment because none of them are directly MIL-STD-1553 compatible.

The MRS operational considerations are shown in figure 6. The operator interface approach for the MRS is similar to the implementation of the U. S. Army IACS program. It utulizes three identical cockpit control displays (CCD) and a dedicated control display unit (CDU) for mission scenario management. The MRS system is also capable of dedicated bus operation and capable of dynamic bus allocation using the external bus controller. The bus loading is less than 10%, with service required at 10 times per second. The critical timing is associated with system reaction to operator keystroke or function selection. The dynamic bus allocation is used to service the navigation

computer unit (NCU) and the control display unit (CDU) for symbolic map display.

The MRS system integrity considerations are shown in figure 7 and incorporate

- redundant controls and displays
- redundant bus
- redundant bus controllers
- partitioning determined from the survivability needs.

Another system which will be described briefly is the U. S. Coast Guard HC-130H program shown in figure 8. This is an operational system already flying in the Kodiak area. You will note a great deal of similarity to the MRS except now the Inertial Navigation System and the h-f filters have been added to the system complement and the navigation computer and its dedicated control display unit have been deleted.

The C-130 is basically a subset of the MRS design. It uses the same system coupler controls, cockpit control displays, and remote readout units. It utilizes a dual bus for redundancy with redundant bus controllers and interface adapter modules for unique equipment complement. The system is used for control and display of the communications and navigation subsystem.

My last example is the GPS/GDM system as shown in figure 9. As can be seen while the bus architecture looks the same much more of the system ties directly to the bus. The GPS/GDM is a test bed currently flying in the Yuma area as part of the GPS evaluation test program. The MIL-STD-1553A functions performed in the GDM included the control/display functions for the receiver, the inertial navigation system (INS), and the high performance antenna assembly. It also served to transfer navigation data to the navigation computer in the form of pseudo range and range rate from the receiver controller, inertial nav data from the INS, altitude data from the altimeter and attitude data from the navigation computer to the high performance antenna assembly. In the GDM system, MIL-STD-1553 was used to provide computer-to-computer data transfer; it transferred time tagged data from the INS and utilized service rates which varied with function up to 50 Hz update rate for the high performance antenna assembly.

In conclusion, while the MIL-STD-1553 applications are very broad, the basic architecture is essentially the same, and the architectural approach can be and is tailored quite easily to meet system needs determined by such consideration as:

- functions performed
- survivability needs
- operator needs, and so forth.

We have concluded that while the architecture is better served by bus controllers as an independent function for large systems, they can be integrated with subsystem hardware for limited subsystems to conserve hardware, weight, and cost. We have also found dynamic bus allocation can be very useful to introduce more sophisticated control requirements when the system demands it.

We believe that we should resist the temptation to add special features to standard equipment for a particular weapon for a particular weapon system. Beyond MIL-STD-1553A, we believe that there is a real need for additional definition, that is, a specification which will provide subcontractor interface control and hardware compatibility and which will actually define the protocol and timing, etc. Of course, in view of the large commitment made to compatible MIL-STD-1553A/B hardware, future definition will have to be upward compatible.

FIGURE 1. IACS ARCHITECTURAL APPROACH

- BASED ON MIL–STD–1553A
- DUAL BUS FOR REDUNDANCY
- REDUNDANT BUS CONTROLLERS
- INTERFACE ADAPTER MODULES USED FOR NON–COMPATIBLE EQUIPMENT COMPLIMENT
- BACK–UP SYSTEM COMPUTERS

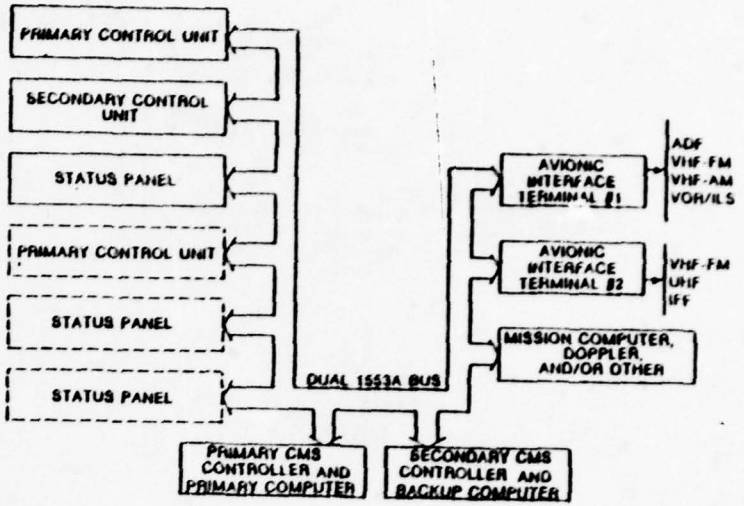


FIGURE 2. IACS SYSTEM COMPOSITION

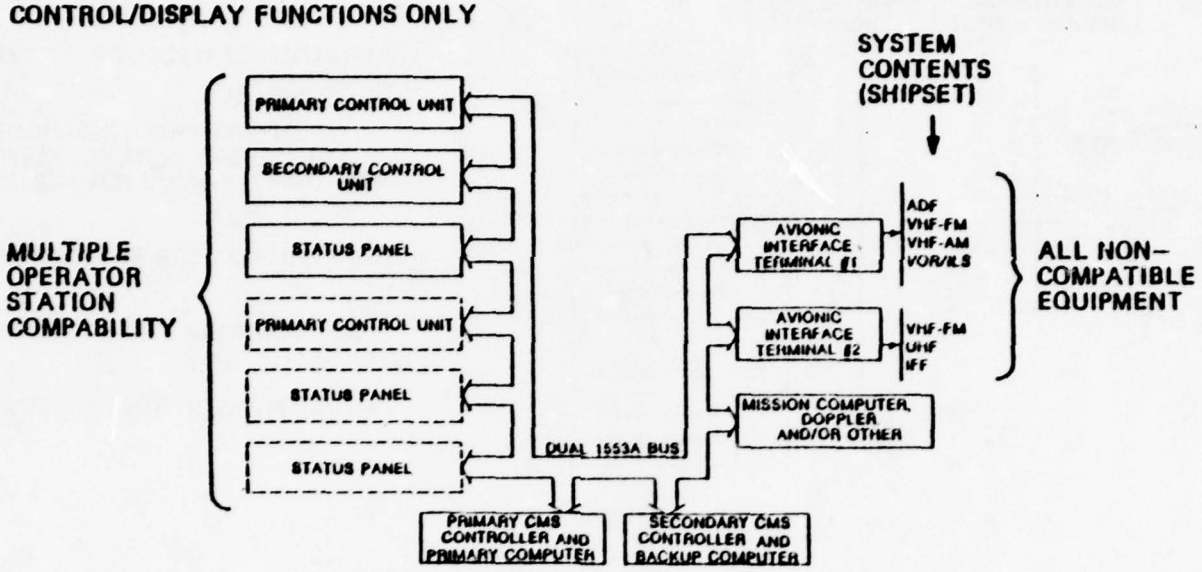


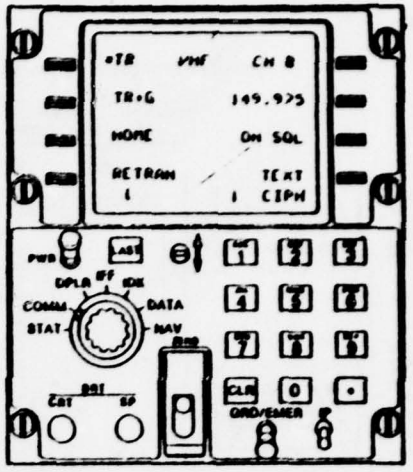
FIGURE 3. IACS OPERATIONAL CONSIDERATIONS

**OPERATOR INTERFACE**

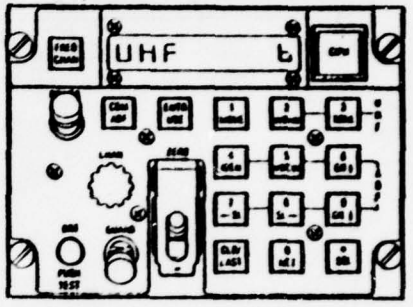
- EXTENSIVE HUMAN FACTORS
- MULTIPURPOSE
- MULTIPAGE DISPLAY
- SECONDARY CONTROL UNIT WITH *REDUCED CAPABILITY*
- MULTIPLE PRIMARY CONTROL UNIT CAPABILITIES

**PERFORMANCE**

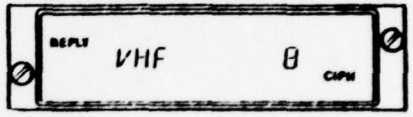
- CAPABLE OF DEDICATED BUS OPERATION
- CAPABLE OF DYNAMIC BUS ALLOCATION USING EXTERNAL BUS CONTROLLER
- BUS LOADING < 10%, 10 TIMES PER SECOND
- CRITICAL TIMING ASSOCIATED WITH SYSTEM REACTION TO OPERATOR KEYSTROKE OR FUNCTION SELECTION




PRIMARY CONTROL UNIT




SECONDARY CONTROL UNIT




STATUS PANEL

FIGURE 4. MRS ARCHITECTURAL APPROACH

- BASED ON MIL–STD–1553A
- SIMILAR TO IMPLEMENTATION OF U. S. ARMY IACS PROGRAM
- DUAL BUS FOR REDUNDANCY
- DYNAMIC BUS ALLOCATION FOR EFFICIENT BUS UTILIZATION
- INTERFACES WITH OTHER DIGITAL STANDARDS IN OFF–THE–SHELF EQUIPMENT
- MAP TRANSFER FROM THE NAVIGATION COMPUTER TO THE CONTROL DISPLAY UNIT

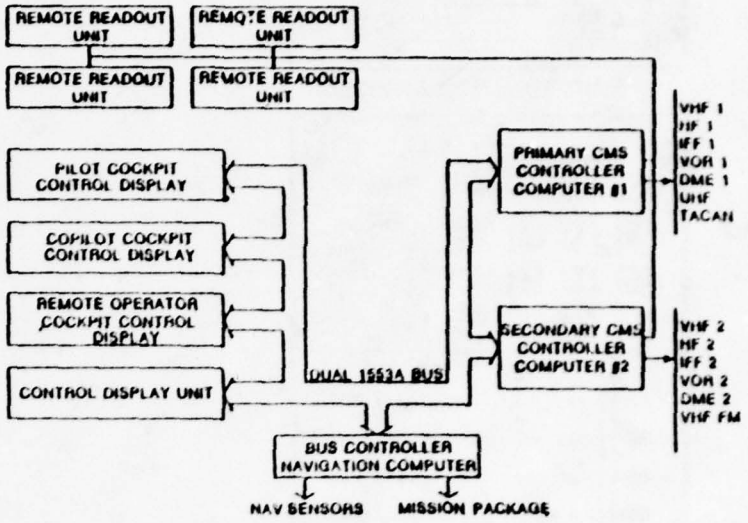


FIGURE 5. MRS SYSTEM COMPOSITION

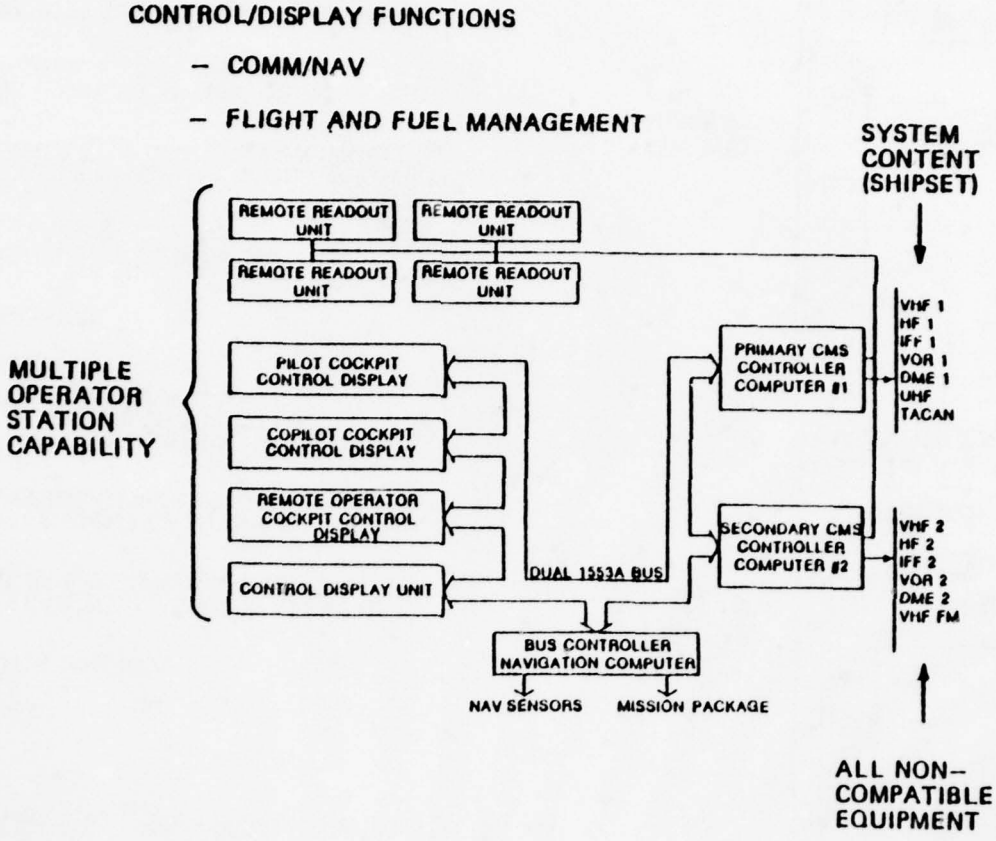


FIGURE 6. MRS OPERATIONAL CONSIDERATIONS

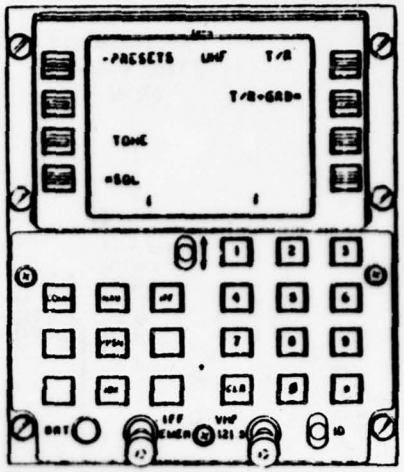
OPERATOR INTERFACE

- SIMILAR TO IMPLEMENTATION OF THE U. S. ARMY IACS PROGRAM
- THREE IDENTICAL COCKPIT CONTROL DISPLAYS (CCD)
- DEDICATED CONTROL DISPLAY UNIT (CDU) FOR MISSION SCENARIO MANAGEMENT

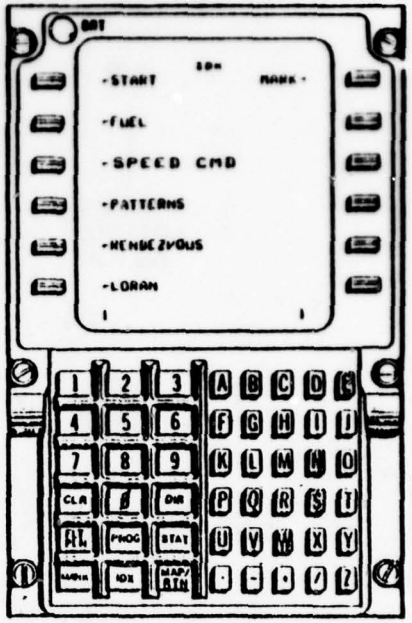
PERFORMANCE

- CAPABLE OF DEDICATED BUS OPERATION
- CAPABLE OF DYNAMIC BUS ALLOCATION USING EXTERNAL BUS CONTROLLER
- BUS LOADING < 10%, 10 TIMES PER SECOND
- CRITICAL TIMING ASSOCIATED WITH SYSTEM REACTION TO OPERATOR KEYSTROKE OR FUNCTION SELECTION
- USES DYNAMIC BUS ALLOCATION TO SERVICE NCU/CDU FOR SYMBOLIC MAP FUNCTION

COCKPIT CONTROL DISPLAY




CONTROL DISPLAY UNIT




REMOTE READOUT UNIT

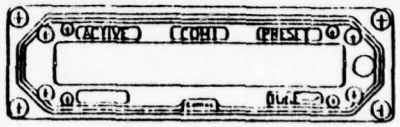


FIGURE 7. MRS SYSTEM INTEGRITY

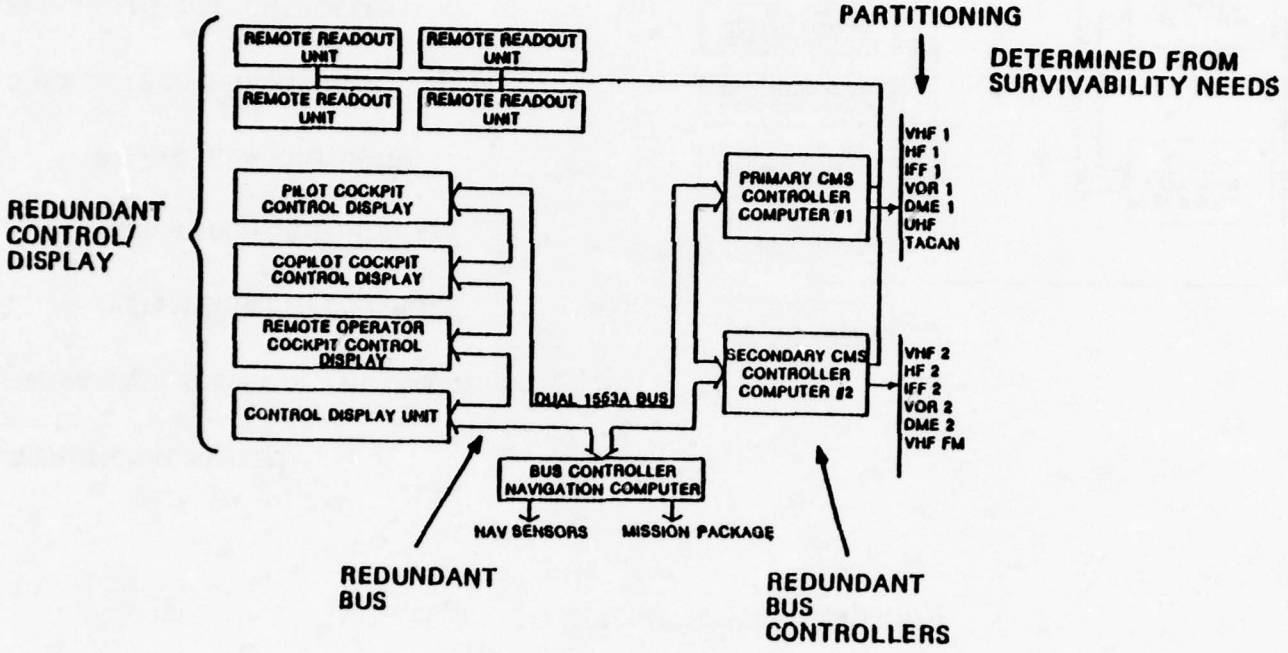


FIGURE 8. C-130 ARCHITECTURAL APPROACH

- OPERATIONAL SYSTEM
- BASICALLY A SUBSET OF THE MRS DESIGN
  - SYSTEM COUPLER CONTROLS
  - COCKPIT CONTROL DISPLAYS
  - REMOTE READOUT UNITS
- DUAL BUS FOR REDUNDANCY
- REDUNDANT BUS CONTROLLER
- INTERFACE ADAPTER MODULES FOR UNIQUE EQUIPMENT COMPLIMENT
- CONTROL AND DISPLAY ONLY

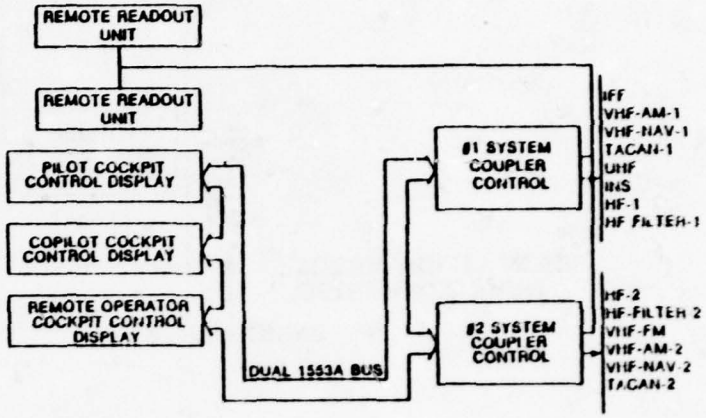
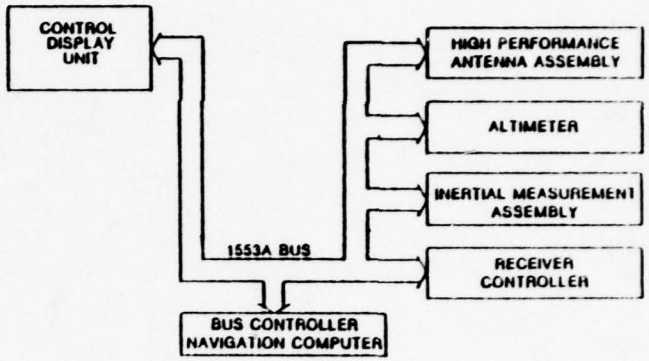


FIGURE 9. GPS/GDM ARCHITECTURAL APPROACH

- A FLYING TEST BED
- MIL-STD-1553A FUNCTIONS
  - CONTROL/DISPLAY
    - RECEIVER
    - INERTIAL NAVIGATION SYSTEM
    - HIGH PERFORMANCE ANTENNA
  - NAVIGATION DATA
    - PSEUDO RANGE AND RANGE RATE
    - INERTIAL NAV DATA
    - ALTIMETER
    - ATTITUDE
- COMPUTER TO COMPUTER DATA TRANSFER
- TIME TAGGED DATA
- SERVICE RATES VARY WITH FUNCTION
- 50 Hz UPDATE RATE FOR THE HIGH PERFORMANCE ANTENNA



# PRECISION ATTACH ENHANCEMENT INTEGRATION

by

Capt Ron Marini, ASD/AER, W-PAFB, OH

(This paper was not submitted)

PRECEDING PAGE BLANK

# MULTIPLEX CONTROL OF AIRCRAFT SUBSYSTEMS

F. E. Funke

Los Angeles Division
Rockwell International
El Segundo, California

## ABSTRACT

The Electrical Multiplex (EMUX) System as developed for the B-1 Bomber represents one of the first extensive applications of time division multiplexing for controlling power to subsystems, data transfer between subsystems, and for control of subsystem operation. This paper reviews the system requirements, configuration, and the operation of the EMUX system including the provision for automatic load management and transfer of data to the on-board integrated test system. The system design approach was to make maximum use of multiplexing where weight or cost effective except where prohibited by safety of flight considerations. The EMUX use of redundancy is described together with fault tolerance concepts and survivability/vulnerability considerations. Interface compatibility with subsystems was assured through standardization and interface control documentation. This was essential as practically all aircraft subsystems utilized EMUX functions and had to be integrated with EMUX. Future improvements were planned for the B-1 Production Program which included redesign to reduce the recurring unit cost of EMUX and the use of solid state power controllers. Various design approaches are under consideration for future tactical and strategic aircraft which would enchance the weight savings and cost effectiveness of multiplex control of aircraft subsystems.

PRECEDING PAGE BLANK

# I. INTRODUCTION

An electrical multiplex system was conceived for the B-1 Bomber early in 1970 because it was realized that weight and volume penalties would result if conventional wiring techniques and relay logic were used to handle the multiplicity of subsystem control functions, the control of electrical power to subsystems, and the transfer of digital data between and within subsystems. Therefore, an extensive use of multiplexing was employed in the aircraft which resulted in the following benefits.

1. Reduced wire harness weight, volume, and fabrication cost.

2. Provided flexibility in control logic permitting system changes with minimal impact on wiring.

3. Provided increased reliability via redundancy where needed.

4. Reduced maintenance and checkout via the on-board checkout system.

The aircraft was partitioned into three multiplex systems. The electrical multiplex system (EMUX), addressed in this paper, is used principally to (1) control power to subsystems, (2) provide Boolean logic processing of subsystem control signals, and (3) provide data transmission channels for aircraft systems. The Central Integrated Test System (CITS) incorporates multiplexing to monitor all aircraft subsystems in flight and on the display/record failed modes of operation, and fault isolate to the LRU level. The Avionics Multiplex (AMUX) integrates the Avionics Control units (computers) with the Navigation and Weapon Delivery Subsystems, Stores Management Subsystem and associated controls and displays for the Offensive and Defensive system operators.

This paper describes the EMUX system requirements, configuration, and operation of the EMUX system followed by the system design approach and resulting applications to the various aircraft subsystems. The improvements and innovations planned for future aircraft multiplex systems is also covered.

## II. SYSTEM REQUIREMENTS

### Functions

In very simple terms, the system requirement for EMUX is to provide a highly reliable, weight, and cost effective substitute for hardwired control, data transfer, and control logic. More specifically, EMUX must perform the following functions:

1. Control and monitor distribution of electrical power to aircraft subsystems.

   To perform this function, EMUX must control and monitor the operation of power control units and monitor the status of the associated circuit breakers.

2. Provide control signals to and between aircraft subsystem LRU's.

   To perform this function, EMUX must sample discrete input signals, process the signals in accordance with pre-programmed Boolean logic, and supply the necessary output signals to subsystem LRU's.

3. Transfer data between subsystems and within subsystems.

   To perform this function, EMUX must act as a common carrier for the transfer of serial-digital data between subsystem LRU's. In addition, EMUX must be able to convert serial-digital data to parallel discrete data and vice versa.

4. Provide automatic load management.

   EMUX must automatically disconnect the aircraft power loads in response to reduction of electrical power generating capability, electrical system ill-health, or aircraft operational modes.

5. Perform self-test.

   EMUX must continuously perform self-test of its own internal functions and automatically shift to redundant internal circuits when necessary.

6. Provide data to the Central Integrated Test System (CITS).

   The EMUX must store all data transmitted within its system in a memory which can be interrogated by the CITS via a serial digital interface. CITS uses the data available from EMUX for in-flight monitoring and ground test of the aircraft subsystems.

7. Execute test commands from CITS.

   During ground readiness testing, CITS configures the aircraft for test and executes many of the required test functions by sending command data to the EMUX through a serial digital interface.

## III. CAPABILITIES

To carry out the foregoing functions, the EMUX system incorporates the following terminals and data buses:

1. Remote terminals.

   The remote terminals provide input and output signal conditioning to interface with the aircraft subsystems. The remote terminals are capable of two-way data transmission with the control boxes (bus controllers) over the multiplex data buses.

2. Control boxes.

   Control boxes (or bus controllers) control the command-response traffic on the data buses and provide the necessary logic processing to develop command data based on the input data received. The Control boxes monitor the self-test data and health of the remote terminals and can take necessary action.

3. CITS interface terminal.

   The CITS (CI) interface terminal stores all input and output traffic transmitted on the data buses so as to make the current data traffic and status of the subsystems using EMUX available to the Central Integrated Test System. The CI boxes also accept serial-digital command data from CITS (during certain ground operational modes) and relay such commands to the control boxes for controlling subsystems and configuring the aircraft for test.

4. Multiplex data buses.

   All communication between the various EMUX LRU's (line replaceable units) are made over redundant data buses using time division multiplex techniques. The data rate is 1 megabit per second using a Manchester II format.

5. Data bus termination.

   Each EMUX terminal interfaces with the data buses through a stub line connected to a data bus termination.

Modes

EMUX operates in either the flight mode or ground mode. In the flight mode, the EMUX carries out all command and monitor functions as programmed, but CITS command functions are locked out. In the ground mode, the CITS command interface is enabled and the EMUX boxes are subject to automatic shutdown if an overtemperature condition developes. This feature was incorporated to make EMUX independent of aircraft cooling restraints during ground testing. The ground mode is interlocked with weight-on-wheels sensors and other signals to insure that this mode cannot be used when the aircraft is taxiing or airborne.

## IV. EMUX SYSTEM DESCRIPTION

EMUX Configuration

The EMUX configuration was based on providing two electrically separate systems (left section and a right section) operating independently and asynchronously with redundancy within each system. Each section uses redundant data buses (primary and secondary) and each EMUX LRU transmits and receives data over both buses.

Each of the data buses consists of a pair of twisted shielded wires, running the length of the aircraft, through which the system components are interconnected. Each EMUX box is connected to the main data bus cables by a termination line stub (2 to 30 feet in length) and a data bus termination. The data bus termination provides stub isolation so that a failure of any stub will not affect the transfer of data between other boxes. The EMUX remote boxes gather and transfer data upon Control box command. For battle damage survivability, the two primary data buses are run down one side of the aircraft and the two secondary buses are run down the other side.

The remote terminal configuration layout was based on meeting the signal quantity requirements in each area of the aircraft so as to minimize the interface wiring, minimize the different types of LRU's required, and to maximize the survivability of the system. The configuration layout for Aircraft 4 (see fig. 1) provides remote terminals distributed from the forward avionics bay to the aft avionics bay. The capability exists for EMUX boxes to be added in the weapon bays on SRAM missile rotary launchers.

Four types of remote terminals were provided to interface with the aircraft subsystems. These terminals primarily differ in the mix of input/output signal conditioning which is provided. The discrete terminals (designated DS-1, DS-2, and DS-3) can only handle discrete on-off or open-close signals. Discrete signals were standardized as 5 volt input/outputs or as switch closure inputs. The discrete-digital (DD) terminals can handle serial-digital signals in addition to the discretes. The EMUX configuration for the Number 4 B-1 aircraft is tabulated in figure 2.

The use of redundant Control boxes, (bus controllers) was selected rather than providing redundancy within a single Control box The Control boxes were centrally located in the Central Avionics Bay. Each Control box has two independent power supplies which are or'd together on the load side. Each power supply receives power from a different source. For battle damage survivability, one Control box of each section was located on the opposite side of the aircraft.

A single CITS Interface (CI) terminal was provided in each EMUX section to provide interfaces with CITS for that section. Since CITS is not a mission critical system, the CI does not contain any redundancy except for the capability to transmit and receive over both data buses.

## V. SYSTEM OPERATION

The EMUX system samples electrical signals during each time frame (nominally approximately 32 milliseconds) and assigns each signal to a time slot (binary bit) within that time frame. All of the sampled signals are then formed into a stream of 24 bit words and transferred over the data link at the rate of 1 megabit per second. This rate permits, in each time frame, transfer of all present B-1 data signals and burden signals (signals required for traffic control and self-test) plus reserve capacity for system growth. This allows all EMUX inputs and outputs to be updated approximately 31 times per second (the frame rate).

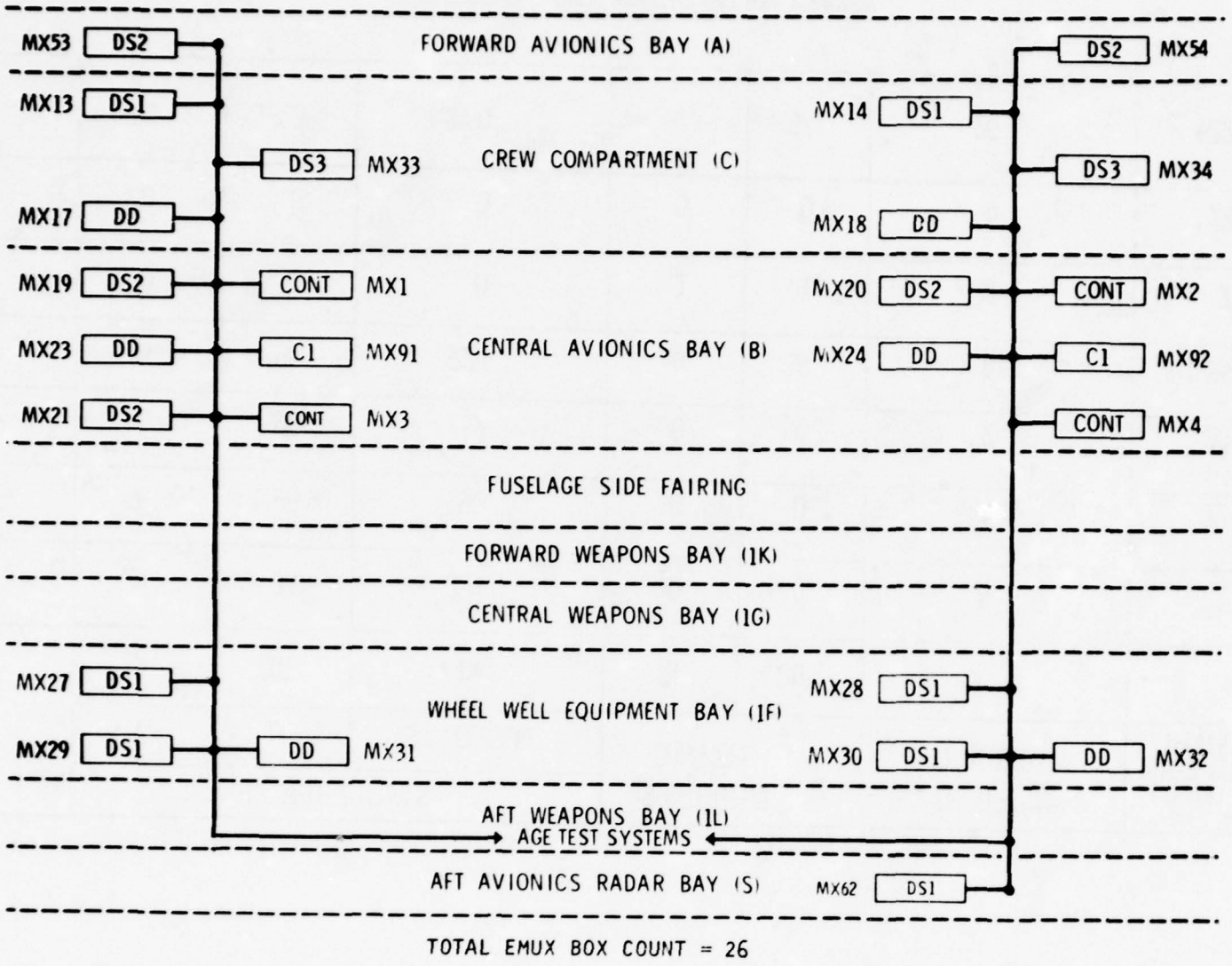



FIGURE 1. EMUX CONFIGURATION, B-1 AIRCRAFT 4

| EMUX BOX TYPE | DISCRETE SIGNALS | | | SER/DIGITAL CHANNELS | | NO. OF TERMINALS PER AIRCRAFT | WEIGHT PER LRU (POUNDS) |
|---|---|---|---|---|---|---|---|
| | 5 V. | DC | SW. CLOSURE | | | | |
| | IN | OUT | IN | IN | OUT | | |
| DS-1 | 24 | 96 | 224 | 0 | 0 | 8 | 22 |
| DS-2 | 56 | 96 | 190 | 0 | 0 | 4 | 22 |
| DS-3 | 0 | 96 | 32 | 0 | 0 | 2 | 20 |
| DD | 24 | 96 | 224 | 16 | 16 | 6 | 33.5 |
| CI | 0 | 0 | 0 | 1 | 1 | 2 | 7.5 |
| CB | 0 | 0 | 0 | 0 | 0 | 4 | 23 |
| TOTAL | 540 | 1920 | 3960 | 97 | 97 | 26 | 612 |

FIGURE 2. EMUX CONFIGURATION SUMMARY
B-1 AIRCRAFT NO. 4

## Operational Sequence

The EMUX system operates as a command-response system in a fixed frame mode. Each of DS and DD boxes is commanded in sequence by the Control box to transmit all stored discrete data to the Control box. The addressed box then transmits a response word, followed by a continuous sequence of data words, until all of the input discrete data has been transmitted. The Control box then addresses the next box, and the transmission sequence is repeated by that box. This process is continued until the Control box has received all the discrete data. After all discrete data has been received, logic processing begins. The Boolean Processor steps sequentially through the instructions stored in its programmable read only memory (PROM), reading each one in turn and performing each operation specified. Using the address portion of the instruction, the Processor interrogates the input memory to obtain the variable (single bit) stored at the bit location associated with the instruction address. Simultaneously, the Processor reads the OP code portion of the instruction and configures its logic accordingly. The input variable obtained from input memory is then processed by the logic and the result is stored in one of two accumulators. The next instruction is then read to obtain the next input variable. It is processed and combined with the previous variable. The result is stored in the accumulator, replacing the previous result. This process continues until an instruction is read that identifies it as the last instruction for a particular equation. At this time, the result stored in the accumulator (the final solution to the equation), is transferred to the Control box output memory. After all equations have been solved and all solutions are stored in the Control box output memory, all results are transmitted to the appropriate remote terminals. All equations are processed each frame.

## Serial Digital Data Transfer

The DD type boxes perform the same functions as the DS boxes, but they also have the added capability of accepting serial digital signals from the aircraft systems. These digital signals must be in a 17-bit per-word return-to-zero (RZ) format. The digital words may be either aircraft generated serial digital signals that will be transferred through EMUX to another part of the aircraft system, or they may be "packed" discretes (discrete signals formatted into serial digital words by aircraft system interface units).

In parallel with the discrete logic processing, the Control box addresses the DD boxes requesting that each box transmit its digital data to the Control box. Each DD box sequences through its digital data input channels and transmits the data without buffering in continuous digital words, until all digital data have been transmitted. The serial digital words are stored in a memory in the Control boxes, and retransmitted to the proper DD boxes within the same frame time.

The Control boxes have the capability to reformat any packed discrete words received into single discrete signals or vice versa.

EMUX Self-Test

The EMUX makes extensive use of built-in-test (BIT) or self test. Each remote box performs self-test in its own operation, can initiate a transfer to its redundant electronic circuits, and the status of the box is sent to the Control boxes in a response word (BIT status word) which is part of the input message to the Control box. The contents of the response word is monitored by CITS via the storage of all data link data transmissions in the CI box memory. The Control boxes can command the remote boxes to switch to their redundant electronic circuits based on lack of response or upon receipt of improper transmissions from the remote boxes.

In each EMUX section, one of two redundant Control boxes assumes active control. The standby box continually verifies that the active box is issuing the correct outputs. If the number of disagreements between the two boxes exceeds 15 for two consecutive frames, the standby or off-line box requests an off-line test. During the off-line test, both boxes run a diagnostic test of their respective Read/Write memory. If the primary or on-line box passes the test, it will continue as the on-line box. If it fails the test, it will remain "off-line" and the standby box will assume control. This "off-line" check requires only about 100 milliseconds and is performed no more frequently than once per minute.

In addition, each Control box performs various self-tests of its own internal functions throughout each frame.

Operation of EMUX is illustrated by a typical application as shown in figure 3 wherein the pilots can control the alternate throttle actuator. In this case, EMUX is a back-up to the hardwired system. The EMUX DS box senses the throttle jog switch actuation, the signals are relayed to the Control box, the Control box sends the appropriate commands to another DS box which controls power to the alternate throttle motors. The Control box develops the appropriate commands depending upon the instructions as programmed for the Boolean processor in the Control box.

The details of the electrical interfaces for the electro-mechanical control of power is illustrated in figure 4. The EMUX remote box outputs a 5 volt command to a solid state relay driver. The relay driver controls the 230 volt AC, 400 hz power to the relay coil. The EMUX remote box monitors the status of the circuit breaker and the relay contacts. This information is used in the Control box in the processing and interlocking of power controller commands. The power controller status can be monitored by AGE during ground operations and by CITS.

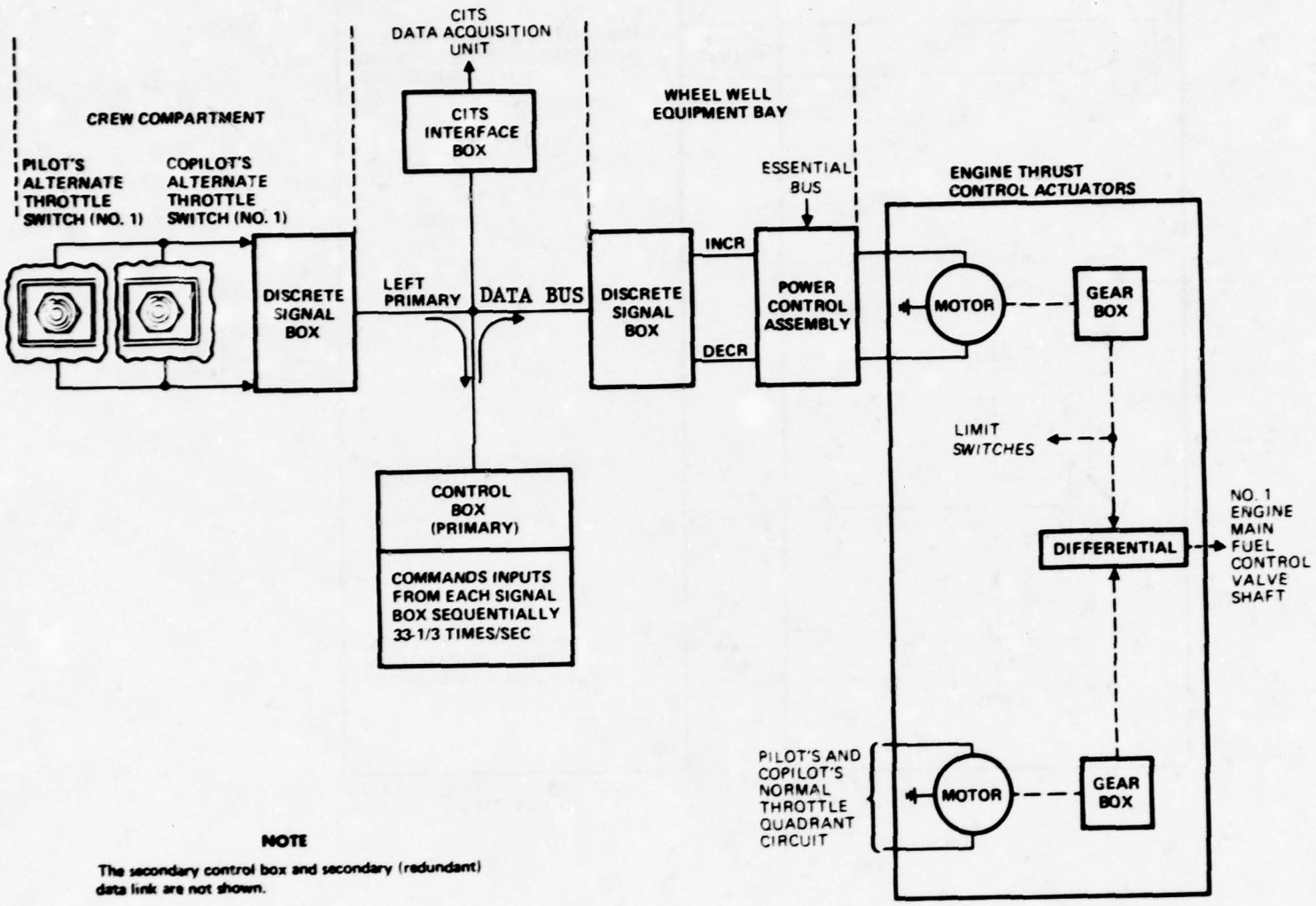



FIGURE 3. TYPICAL APPLICATION OF EMUX

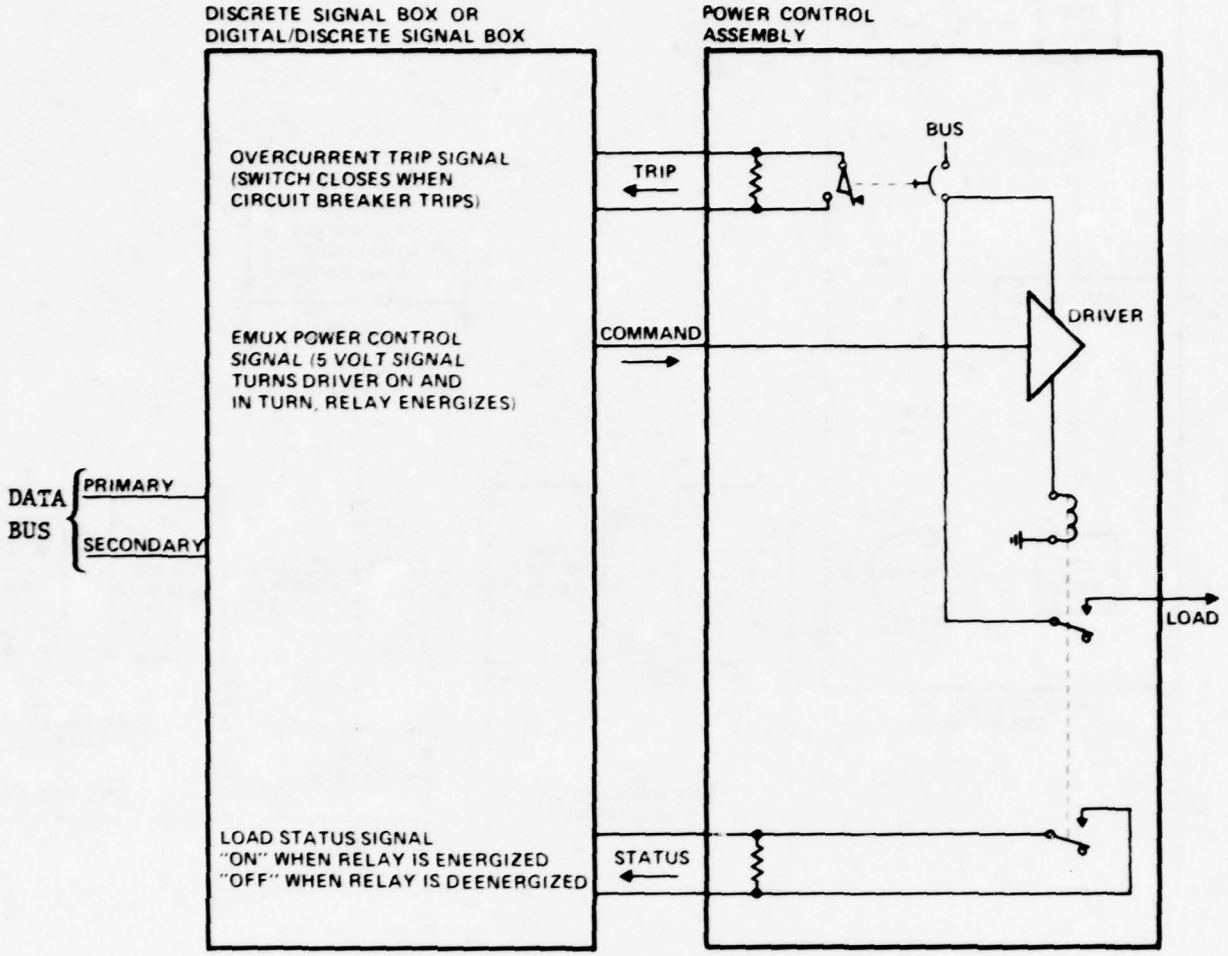



FIGURE 4. EMUX INTERFACE WITH ELECTRO-MECHANICAL POWER CONTROLLERS

Programming

To enable the EMUX to properly carry out the required logic and power control functions, the control memory in the Control box must be programmed using appropriate software. The Control box sequentially reads instructions stored in its control memory and sequentially performs the functions defined in each instruction. Simply stated, it reads an instruction, performs the function, and then reads the next instruction. The control memory is a programmable read only memory (PROM).

The PROM, when properly programmed, contains the following information:

A. Sequence Control - This information defines the sequences in which the Control box is to collect and transmit discrete and serial digital data, performs self-test, starts and stops processing, and responds to executive or external subroutine commands. In addition, it contains serial digital routing instructions.

B. Processing - These instructions define the functions required to solve all Boolean and Time Delay equations, the sequence of solutions, and the memory location of input and output signals.

C. Load Management - These instructions define the Load Management routines.

The PROM is physically programmed by external AGE. Up to 11,264 twenty-one bit words must be programmed into the PROM. Obviously, any manual programming method would involve many man-hours and would be subject to a considerable degree of error. To accommodate economic and proficient programming, an automatic device termed the "Memory Loader Verifier" is utilized. This device is capable of accepting programming information stored on tape, and using this information, rapidly programs each PROM card. In addition, it is capable of testing the PROM to verify its programmed content.

The Memory Loader Verifier requires input data stored on tape. This tape is prepared using an IBM-360 Compiler Program. The Compiler Program accepts raw data decks and transposes this data into a set of machine language instructions, stored on tape, to be used by the Memory Loader Verifier to physically program the Control box PROM cards. The Compiler Program is IBM-360 compatible and is written in Fortran IV language.

## Load Management

Load management is a function of the Control box which modifies the normal control of the electrical loads (through power controllers) as a function of aircraft configuration and operational mode. This function of the Control box protects the primary power generation system from becoming overloaded in the event of a multiple electrical generator failure, so that loads will be maintained within the generator and aircraft cooling system capacity. Load management is also used during ground operations, to assist in configuring the aircraft during test. Load management is accomplished automatically by inhibiting or powering down applicable aircraft systems and/or equipment as dictated by a predetermined schedule for a particular load management mode.

The Control boxes receive and store appropriate discrete priority reference signals provided by various systems. Generator load contractor, bus tie contractor, auxiliary power unit (APU), and engine status are typical examples of priority reference signals. Using the priority reference signals, the Control boxes in each EMUX section logically select one of twelve available load management schedules stored in each Control box memory. Each schedule consists of 720 power control priority bits (ones or zeros) in a read-only memory (ROM). The bits enable or disable the operation of 720 power controllers in each EMUX section. The Control boxes output commands to power controllers as required to inhibit or power down applicable systems and/or equipment. The following load management modes are provided:

### Refuel/Defuel Mode

The refuel/defuel mode (mode 1) provides limited electrical power and is used during ground refueling and defueling and during other compatible limited load functions on the ground.

### Alert Mode

In the alert mode (mode 2), both APU's are operating and bled and at least two generators (one driven by each APU) are on the line. Electrical power for alert equipment is limited to about 46 kva and reduced to about 14 kva during a four-engine start. The reduction in power is required for simultaneous four-engine start without exceeding the power capability of the APU's.

One-Engine-Driven Generator Mode

The one-engine-driven generator mode (mode 3) provides power for loads required to maintain a safe aircraft with mission strike capability when only one of three engine-driven generators remains on line.

All-Up Mode

The all-up or normal mode (mode 4) provides power for all electrical loads on the aircraft. This mode is enabled automatically if no other mode is enabled. When airborne, mode 4 requires at least two engine-driven generators to be on the line. This is the normal flight mode wherein all loads are supplied with power on demand.

All-Up Maintenance (CITS) Mode

The all-up maintenance (CITS) mode (mode 5) is enabled when the EMUX/CITS ground maintenance interface has been enabled. There are no specific power supply requirements for this mode. External power or any combination of generators (engine number or APU-driven) can be utilized. In this mode, CITS controls the total load so as to limit the load to be within the power source capability.

Proximity Switch Test Mode

In this mode, CITS can perform tests of proximity switches with all systems and equipment powered down except the proximity switch electronics units, CITS, and the aircraft environmental cooling system.

Ground Operating Mode

The ground operations mode (mode 8) provides an override of the automatic load management mode selection and is required for post-flight drift testing of the Inertial Measurement Unit (IMU), and to simplify the procedure for transitioning from ground cooling to internal cooling of equipment bays.

## VI. SYSTEM DESIGN PHILOSOPHY AND REQUIREMENTS

The B-1 EMUX system design is based on using a sampled data multiplex control system to provide the control of as many aircraft functions as possible within the following constraints.

1. Safety of flight functions shall be hardwired or be provided with a hardwired or electro-mechanical backup if multiplexed.

2. The use of multiplexing will reduce wiring or is required to provide the necessary control logic.

The system philosophy was that EMUX was essential to mission success but the aircraft could be flown and returned to base if EMUX suffered a major or complete failure. Since EMUX was one of the first extensive uses of multiplexing, the decisions to multiplex or hardwire received much attention as there were reservations about using such a new system. The Customer (U.S. Air Force) formed a review panel to evaluate the overall mix of hardwire and multiplex. The chart shown in figure 5 is representative of the mix of control techniques that was achieved. As shown, there were definite reasons why the use of multiplexing was not used for control of certain functions. In general, all aircraft functions were multiplexed except for the following:

1. Short wire runs and wherein Boolean logic control operations were not required.

2. Analog or servo type signals (the number of signals to be handled did not justify the resulting weight and complexity penalties in EMUX).

3. Signals required when EMUX would be inoperative (pre-start up).

4. Back-up or emergency functions for safety of flight.

5. Critical nuclear or conventional weapon signals.

The general approach was to use EMUX as much as possible and in the most cost effective manner. Even where signal traffic was localized, EMUX was used when the logic processing capabilities and redundancy was required or when the Central Integrated Test System (CITS) required access to the data via EMUX. EMUX is unique relative to other on-board systems including CITS in that electrical switch closures (open or closed) can be sensed as a source of data.

| SUBSYSTEM | % EMUX | % HARDWIRE | WHY HARDWIRED |
|---|---|---|---|
| FIRE PROTECTION | 5 | 95 | EMUX NOT ON DURING APU STARTUP |
| AUXILIARY POWER UNIT (APU) | 20 | 80 | EMUX NOT ON DURING APU STARTUP |
| ACCESSORY DRIVE | 90 | 10 | EMUX NOT ON DURING APU STARTUP |
| ELECTRIC POWER GEN & DISTR | 50 | 50 | BASIC POWER DISTRIBUTION |
| ENGINES | 84 | 16 | SYNCHROS, VALVES, ANALOG |
| AIR INDUCTION CONT | 50 | 50 | SHORT WIRE RUNS |
| FUEL | 98 | 2 | HIGH FREQUENCY SIGNALS |
| ENVIRONMENTAL CONT & PROT | 95 | 5 | RAM-AIR SCOOP, ALTERNATE |
| LANDING GEAR | 90 | 10 | EMERGENCY EXTEND |
| HYDRAULIC INDICATION | 50 | 50 | ANALOG |
| FLIGHT INSTRUMENTS | 10 | 90 | ANALOG |
| FLIGHT CONT | 20 | 80 | SERVOS & ANALOG |
| WEAPON CONT (CONVENTIONAL) | 90 | 10 | WEAPON DISARM |
| WEAPON CONT (NUCLEAR) | 20 | 80 | AIR FORCE DIRECTIVE |
| CREW ESCAPE | 2 | 98 | EMUX INOPERATIVE |
| CREW ACCOMMODATIONS | 67 | 33 | SHORT RUNS |
| LIFE SUPPORT | 75 | 25 | ANALOG |
| LIGHTING | 67 | 33 | ANALOG |
| CITS POWER CONT | 0 | 100 | |
| AVIONICS POWER CONT | 95 | 5 | SHORT WIRE RUNS |

FIGURE 5. EMUX VERSUS HARDWIRED SUBSYSTEM FUNCTIONS

## Redundancy

The EMUX system is capable of meeting the subsystem reliability requirements through the use of two independent EMUX sections (systems), the use of redundant data buses, the use of redundant electronics in each of the EMUX remote boxes, and the use of redundant input/output channels to interface with subsystems or power controllers.

The subsystem design engineer had many options to choose from and could establish the degree of redundancy provided for any given requirement. In addition, the EMUX software provided the use of Boolean processing wherein the probability of false operation could be minimized by the use of voting logic or other logical permutations as a safeguard.

Each EMUX section uses two data buses for transmission and reception of data. Each EMUX LRU transmits and receives data over both buses, but the data used from either bus is selected after validity, sync, parity, and format checks.

Each EMUX remote terminal is provided with fully redundant electronics except for the input and output buffers which are common to both electronic blocks. The EMUX box will shift to the redundant block of circuits as a result of an internal self-test determination of upon command by the Control boxs. Figure 6 is a diagram of the redundancy of a typical remote terminal. Each remote box receives two independent sources of power and incorporates two power supplies. One power supply is dedicated to the primary block and the other to the secondary block.

Redundancy was provided for subsystem interfaces, when required, by using EMUX inputs and outputs from different EMUX boxes. When redundant power control was required, various configurations of parallel and series relay drivers were used to protect against power "fail on" or "fail off."

## Fault Tolerant Concepts

The aircraft was made fault tolerant to EMUX failures by the use of the following techniques.

1. Use of both EMUX sections.

   An example is the weapon bay doors. If the left EMUX does not carry out the door commands, the alternate door system is operated by right EMUX.

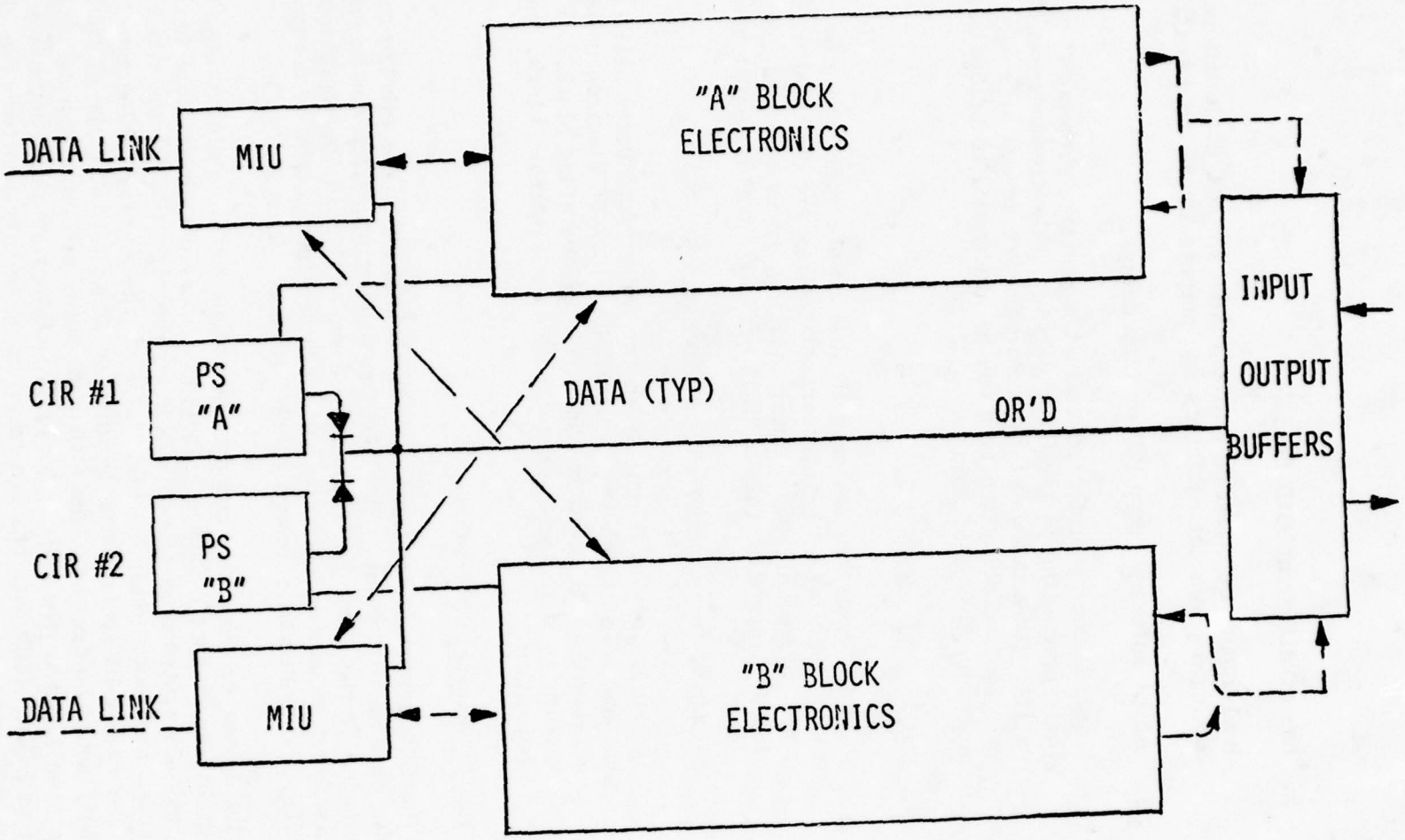



FIGURE 6. EMUX REMOTE TERMINAL ELECTRONIC CIRCUIT
BLOCK REDUNDANCY

2. Use of different EMUX boxes.

   Dual inputs or commands can be routed through different boxes making a given set of inputs and outputs immune to a box failure.

3. Use of software "Pump-Up" or time delays.

   Commands can be protected against mementary "drop-outs" by suitable logic processing in the Control boxes. The commands can be held until a given number of input samples are zeroed or a slow to release time delay function can be programmed to bridge across momentary losses of data.

4. Use of fail-safe modes.

   The EMUX boxes are designed to fail to the "off" state or shut down. Therefore, the aircraft subsystems are designed to accommodate a fail off condition. Certain critical power circuits use normally - closed relays such that a "fail off" will not interrupt power.

5. Switching to redundant circuit blocks.

   The built-in-test circuitry in the EMUX remote boxes will cause the boxes to switch to the alternate block of electronics when a failure is detected in the initially operating block. If a failure is subsequently detected in the alternate block, the box shuts down.

Survivability/Vulnerability

The EMUX system is hardened against the radiation and electro-magnetic effects of a nuclear event. The EMUX Control box operation may be momentarily interrupted by the event but the remote boxes will hold their previous state outputs for 2 seconds. Since the boxes are designed to not be damaged by EMP/TREE, the system will resume operation after the event.

The layout of the boxes and data buses was based on minimizing EMUX susceptibility to battle damage. The two primary data buses (for both sections) are routed down the left side of the aircraft and the two secondary data buses are routed down the right side of the aircraft. The two Control boxes for each EMUX section are deployed so as to minimize the probability that both would be damaged. The EMUX data buses are protected against shorts on the stub lines to the EMUX boxes by transformer resistor network terminations. Each EMUX data bus can tolerate up to five shorted stub lines without suffering excessive degradation of the signal.

## MIL-STD-1553 Compatibility

The EMUX system was conceived in the 1970-71 time period prior to the issuance of any military standard for aircraft multiplexing. However, the EMUX system influenced the content of MIL-STD-1553 (Aircraft Internal Time Division, Multiplex Data Bus System) as was initially released in August 1973. Engineers associated with EMUX served on the Society of Automative Engineers (SAE) Subcommittee A-2K which assisted the Air Force in formulating MIL-STD-1553. As a result, the EMUX system is largely in agreement with the MIL-STD relative to transmission bit rate, signal formats, electrical characteristics and command-response protocol. The EMUX is not compatible with MIL-STD-1553 in that EMUX uses 24 bit words versus 20 bit words for 1553, and the command/status words utilize a slightly different word format. MIL-STD-1553 revision "B" (scheduled for release in 1978) allows certain more sophisticated modes of operation such as the simultaneous broadcast to all remote terminals. The revised MIL-STD gives the system designer considerable latitude in meeting a system requirement but does standardize on data bus characteristics, transmission formats, protocol, mode codes, status codes, and electrical interfaces. Compliance with the standard may not result in interchangeable hardware but should maximize inter-operability.

## Interface Compatibility

One of the significant factors which contributed to the successful integration of EMUX on the B-1 was standardization of interface signals which in turn resulted in interface compatibility. An internal Rockwell International (RI) engineering specification was written which provided the electrical parameters for discrete and serial digital signals and signal conditioners. This document controlled voltage levels, signal formats, timing logic level, etc. This specification was invoked on practically all suppliers of hardware for the various subsystems and was adhered to by EMUX and CITS. As a result, the number of interface problems were minimized.

The compatibility between EMUX and the equipment supplied by the associate contractors for the offensive and defensive avionics was assured by specifying in detail the interface parameters in the avionics interface control documents (ICD). In addition, tests were conducted in Boeing's System Development Laboratory (SDL) wherein a partial EMUX system (Control box and two remote boxes) were integrated with Boeing's Interface Adapter Units (IAU) and other avionics controls. This testing assured that the two asynchronous multiplex systems (EMUX and AMUX) could reliably communicate with each other.

## VII. SYSTEM APPLICATIONS

EMUX was applied so extensively throughout the B-1 that practically all aircraft systems were integrated with EMUX. Figure 7 contains a list of the systems and the functions associated with those systems that were handled by EMUX.

### Avionics Power Control and Stores Management

One of the more extensive applications of EMUX is involved in the control and statusing of all electrical power to the offensive and defensive avionics together with stores management. The EMUX interfaces with the Avionics Control Unit (ACU) complex via serial digital interfaces with AMUX. (See figure 8.) The control and status traffic between the two multiplex systems is handled over two-way serial-digital traffic between the two multiplex systems is handled over two-way serial-digital channels between the avionics Interface Adapter Unit (IAU) and an EMUX remote terminal (DD box). The two way channels are redundant and cross-strapped for reliability. The power control commands are "packed discretes" in that each 16 bits plus parity word contains 16 different discrete on-off commands or groups of bits as used to control weapon bay door position or rotary launcher positioning. In turn, the status of power to the avionics and weapon stores together with weapons bay door and rotary launcher position is transmitted to the ACU via EMUX and AMUX. The end result is that the Weapon Delivery ACU, Guidance Navigation ACU, and the Defensive ACU can control power to the avionics boxes, SRAM missiles, ECM equipment and facilate the launch of the missiles through control of launcher rotation and weapon bay doors.

## VIII. FUTURE IMPROVEMENTS AND INNOVATIONS

As a pioneer system in the art of aircraft multiplex control and data transfer, EMUX was very successful and cost effective. However, a second generation design was planned for implementation on the Production B-1 Program prior to the Presidential decision to terminate the program. The primary purpose of the redesign was to reduce the recurring hardware costs by incorporation of various electrical and mechanical improvements. The major electrical improvement was to convert the many types of hybrid integrated circuits to monolithic circuits which would result in reduced unit cost and higher reliability. Additional volume was made available in the boxes for the power supplies which allowed the power supplies to be repackaged in a more producible and economical configuration. Additional instruction memory was provided for the Control boxes which would facilitate future growth in the use of EMUX. Miscellaneous chassis improvements were made such as using castings instead of machined components which would further reduce production costs.

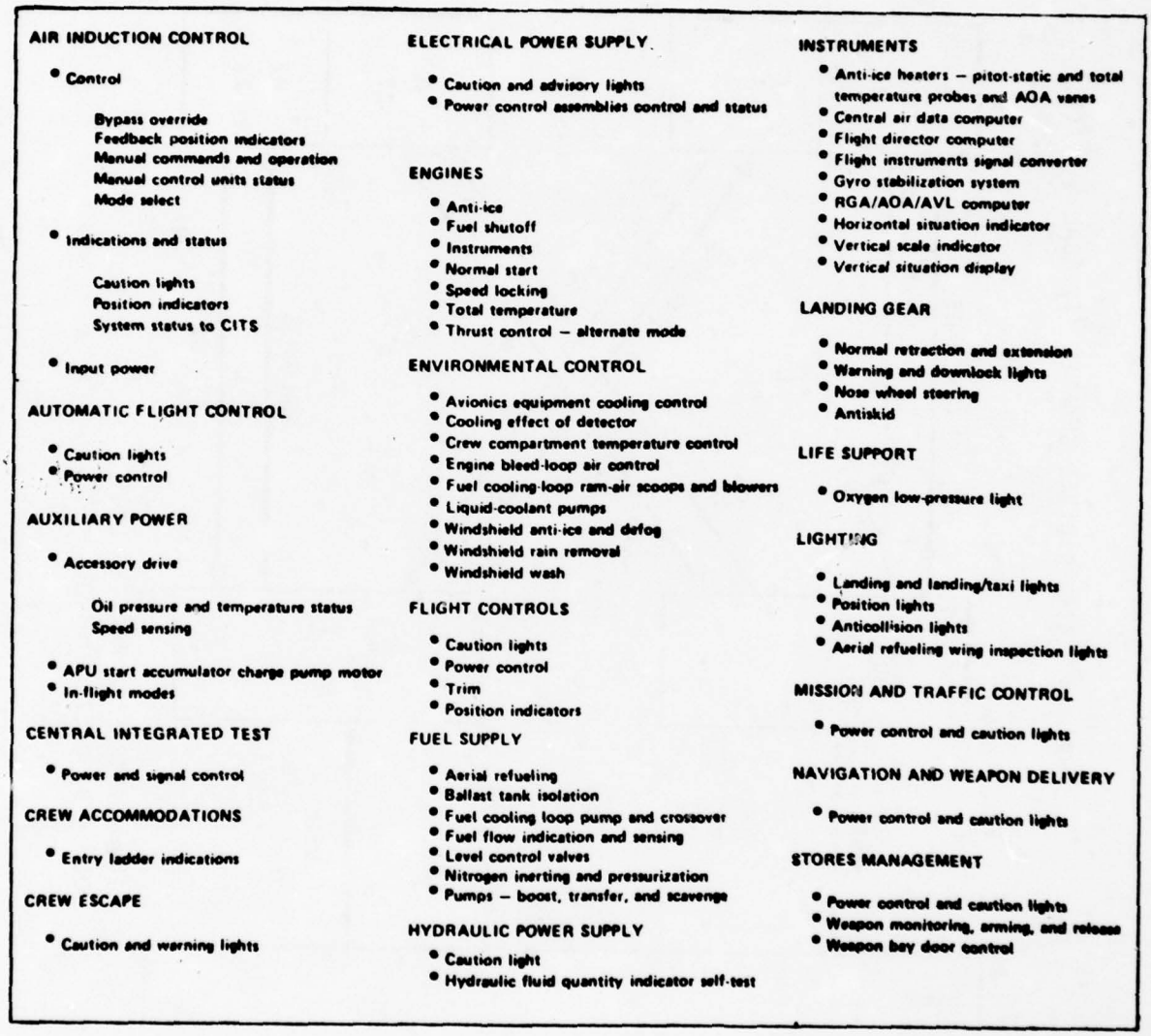



FIGURE 7. SYSTEMS INTEGRATED WITH EMUX

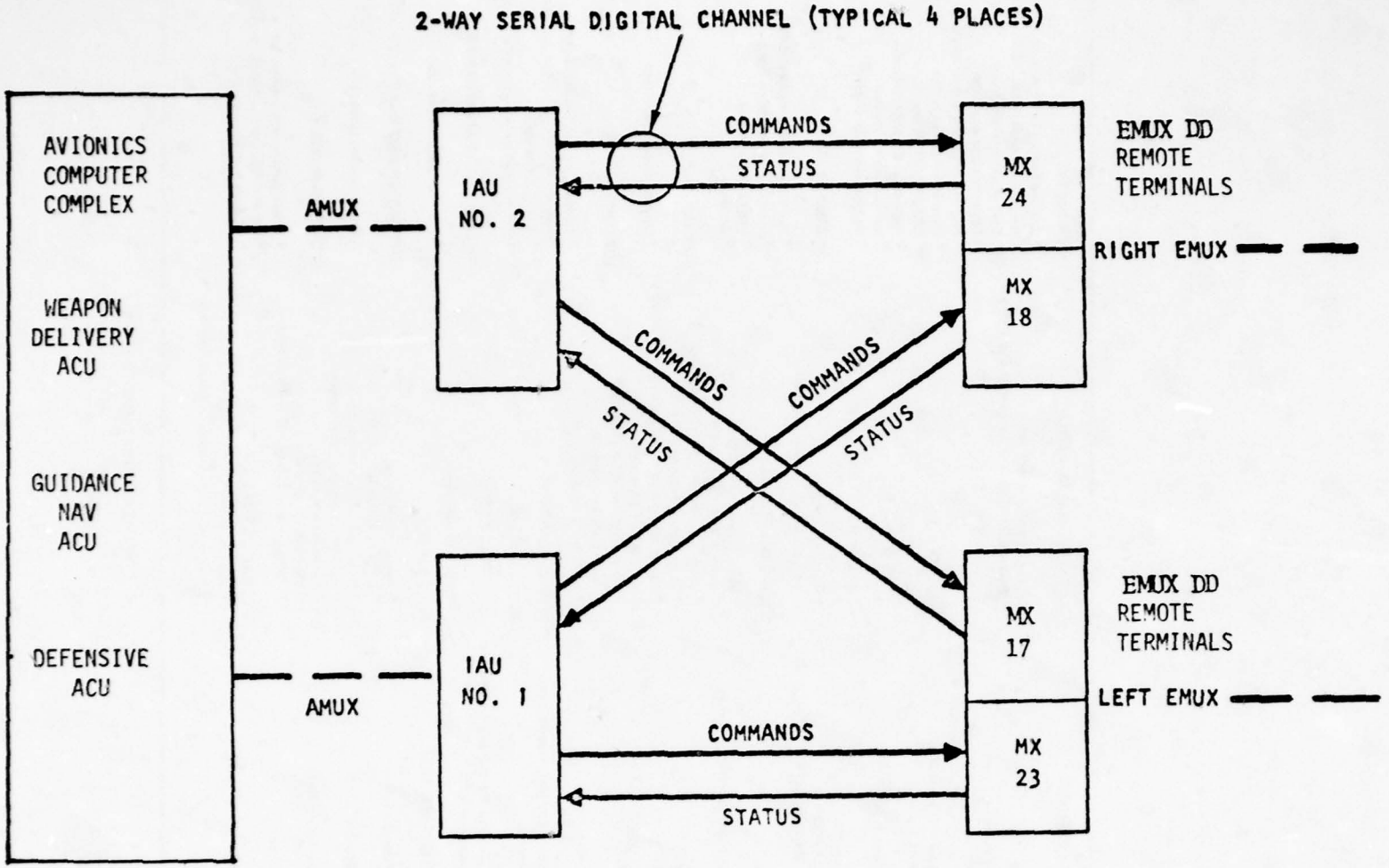



Figure 8. AMUX/EMUX Serial Digital Interfaces

## Solid State Power Control

The four research B-1 aircraft used only electromechanical circuit breakers and relays, however, the EMUX system was designed to be used ultimately to control solid state power controllers. For the production aircraft, the Autonetics Group of RI developed a solid state power assembly (SSPCA) and three types of internal circuit modules. The SSPCA is designed to be a line replaceable unit (LRU) whereas the internal modules will be shop replaceable units (SRU's). Figure 9 illustrates the planned configuration of the SSPCA.

The power control modules are printed circuit boards, each of which contains four power control circuits. The other two types of modules are the status/isolation module used to provide isolated status signals, and the logic modules used to provide "or" - "and" gating of command signals to the power control modules.

## Solid State Power Controller (SSPC)

The SSPC is a polarized, bistable, solid-state device having an output circuit that can be actuated by a low level polarized signal to the input circuit. The output circuit presents a high impedance to the flow of current in its "off" or tripped condition and low impedance in its "on" condition.

The primary functions of the SSPC are (1) to provide 230 VAC, 400 hz electrical power to a specific aircraft load upon receipt of a 5-volt DC control signal, and, (2) to provide overload protection to the aircraft wiring in the event of an overcurrent situation resulting from an electrical overload. Secondary functions are (1) to provide status information of the SSPC to EMUX and, (2) to provide overload trip information to EMUX. (See figure 10.)

The current sense and trip timing circuit tracks the load current by monitoring the voltage drop across a fixed resistor in series with the load. It provides trip signals to the control logic based on predetermined current versus time profiles. The voltage sense circuit tracks the voltage across a resistor divider network at the SSPC output. From this it generates the status signal for the control logic. Voltage sensing is used in lieu of current sensing for improved fault isolation.

The control logic circuit utilizes the trip signals, status signal, and the external control signal as well as zero current crossing and zero voltage crossing signals. It verfies that the input command is valid and controls

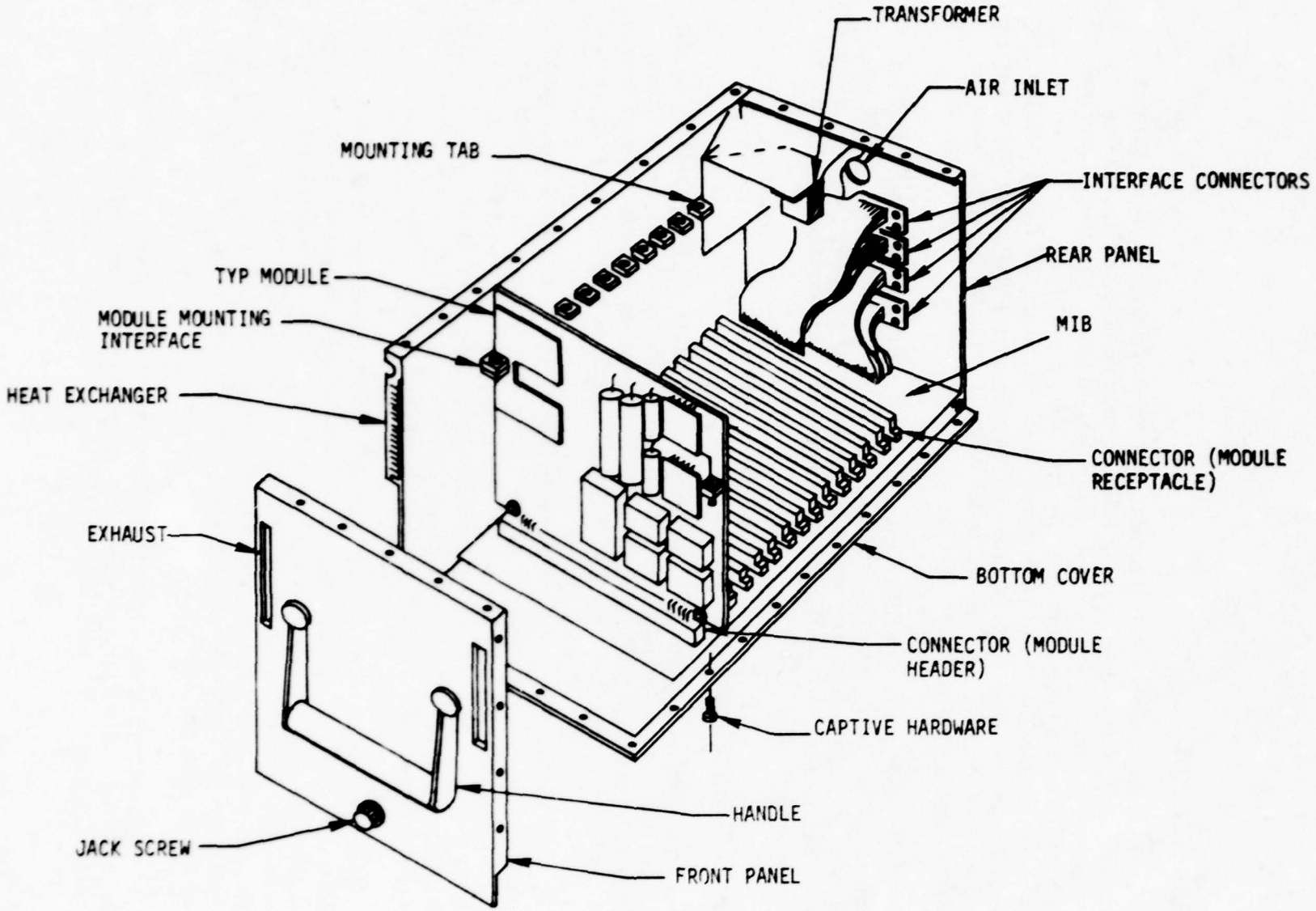



FIGURE 9. SOLID STATE POWER CONTROLLER ASSEMBLY (SSPCA)

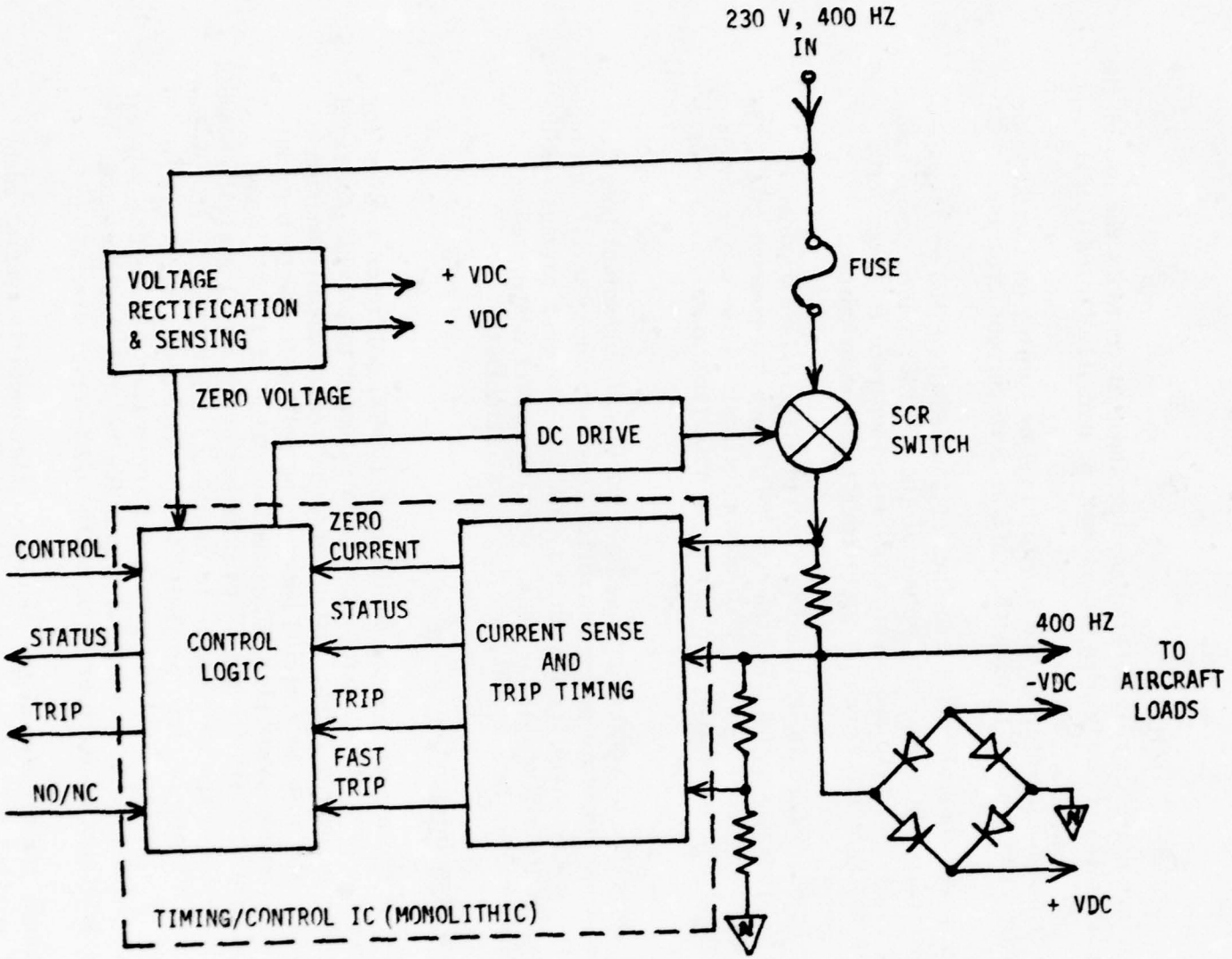



FIGURE 10. SOLID-STATE POWER CONTROLLER (SSPC)
FUNCTIONAL DIAGRAM

the SCR switch drive circuit. Using another input it controls the use of the SSPC in either the normally open (N.O.) mode or normally closed (N.C.) mode.

Electromagnetic interference is minimized by turning on the SCR's at zero voltage crossing and turning them off at zero current crossing.

Each power control module consists of four SSPC's rated at 2 amperes RMS, all wired to the same phase of one of the 230 VAC, 400 Hz power buses. The control, status, trip and power outputs are separate for each SSPC. Individual failsafe functions are provided for the power outputs

The standard SSPC is designed to functionally replace a normally - open relay. This means that the power is "off" when the command is 0-2 VDC, and "on" when energized by a 3-6 VDC, command signal. Some SSPC's can be jumpered to either a normally-open (N.O.) or normally-closed (N.C.) configuration.

The objective of converting specific B-1 electromechanical power controllers to solid state power controllers was to reduce the life cycle costs of the Power Controller Assemblies (PCA's). The cost savings would be realized in lower acquisition costs, reduced support costs, reduced volume requirements, less weight, and improved reliability.

Increased System Integration

Based on the experience gained in the B-1 EMUX, multiplex systems for future strategic or tactical aircraft will be extended to provide increased integration of avionics, flight instruments, and power control functions. In particular, the interface wiring between the multiplex remote terminals and the power control assemblies (PCA) can be reduced if the PCA's can communicate directly with the multiplex data bus or if serial digital channels are provided between the EMUX remote terminals and the PCA's. In the present system, the PCA's are in close proximity to the EMUX remote terminals but three pairs of wires are used for each power controller. Therefore, a great multiplicity of wires can be replaced with a few twisted shielded pairs in either the multiplex approach or in using the EMUX serial digital channels.

Although, the present master caution display panel is controlled by EMUX, each parallel discrete signal requires a wire pair. By integrating a multiplex terminal in the master caution panel, the wiring can be greatly reduced in the instrument panel area which is extremely constricted.

With the increased confidence which has been gained through the B-1 test program, multiplex techniques can be extended to other systems and functions which were not initially candidates for multiplexing. An example, is the power generation and bus distribution system. Whereas, EMUX only provided load management and status of this system, the integration of generator controls and system controls with the EMUX multiplex capability is under consideration and study.

On a large aircraft such as the B-1, the use of three major multiplex systems was justified based on the hardware state-of-the-art, acceptance of multiplexing at the inception of the program, and number of signals to be handled. The contractual partitioning of the program was such that an associate contractor integrated the avionics and developed a separate multiplex system (AMUX). In the future, and particularly for smaller aircraft, the use of a single multiplex system would be desirable. The operational experience gained with EMUX indicates that provisions for redundancy can be reduced such as using a single EMUX multiplex section in lieu of two independent sections.

MULTIPLEX DATA BUS CONFERENCE

10-12 OCTOBER 1978

DAYTON, OHIO

SESSION III

NEW DATA BUS CONCEPTS - PART I

MODERATOR: Lt Col John Marciniak
AFSC/XRF



PRECEDING PAGE BLANK

# ULAIDS - 1553 ARCHITECTURE

# with

# DYNAMIC BUS ALLOCATION

by

Louis Ciasulli
Patrick Henderson

Conrac Corporation
West Caldwell, New Jersey

## ABSTRACT

This paper describes the development and operation of a universal multiplex which satisfies the requirements of MIL-STD-1553A, MDC 3818, and MIL-G-85013 and is consistent with the proposed MIL-STD-1553B.

First, background information is provided on CONRAC programs which led to the development of the multiplex. Following this, a detailed description of a universal bus controller interface unit (BCIU) is given. The paper closes with descriptions of three programs which utilize this BCIU.



PRECEDING PAGE BLANK

## 1. INTRODUCTION

The multiplex architecture described in this paper utilizes the knowledge and experience gained in many years of designing, developing, and producing MIL-SPEC avionics multiplex bus systems. The following is a list of programs involving mux bus technology:

- US Navy Universal Locator-Airborne Integrated Data System
- NASA Space Shuttle Engine Interface Unit
- US Air Force F-15 Signal Data Recording Set
- US Navy F-18 Stores Management Set
- NASA Space Shuttle Ground Command Interface Logic Unit
- US Navy LAMPS Mux Bus Tester
- US Navy F-18 Communications System Control Set

As a result of the ULAIDS program, which required a distributed processing system, Conrac designed a generalized architecture which could be used for both polled contention and command response requirements. This architecture, designated Integrated Data Acquisition and Display System (IDADS) has the following design features:

- Communications between multiplex terminals is either distributed or command response (Hybrid System)
- Each multiplex terminal is self-contained and intelligent and is capable of initiating or responding to transmissions from/to other terminals

- Each terminal is software controlled and except for I/O, is similar from a hardware standpoint, thus achieving commonality and standardization among the various multiplex terminals in a system

- The varying requirements of a user (data terminal) are configured into the system by I/O cards connected to the microprocessor bus in the multiplex terminal. These data terminals are universal and include displays, data entry devices, analog inputs, digital inputs, avionics, weapon systems

- IDADS is flexible and capable of expansion. The system can be configured with any number of data stations.

## 2. IDADS DESCRIPTION

IDADS is an integrated MIL-STD-1553A data system using distributed processing and communications. In the IDADS system, each user is an intelligent terminal capable of direct communications to/from any other terminal.

Figure 1 is a block diagram of Conrac's IDADS approach. Communications between each of the terminals is accomplished using a polled MIL-STD-1553A (redundant) bus technique which makes use of a Conrac-developed floating bus controller. The individual terminals are internally self-sufficient in that they are capable of processing any data that is collected and initiating communications. The work load for the system is distributed among the various subsystems and thus the overall data processing speed is increased.

Each station connected to the multiplex bus consists of a multiplex terminal (communications interface) and a data terminal. The multiplex terminal is a standard hardware configuration containing a bus interface unit, microprocessor, and an I/O controller. The I/O controller is designed to interface with the data terminal. The system allows interfacing of the multiplex terminal with any type of data terminal under software control. The data terminal can be a display subsystem, tape recorder, data entry device analog signal conditioner, avionic subsystems, etc.

The floating controller provides for communications on the bus using MIL-STD-1553A protocol. In this technique, bus control is handed to the next designated station (under software control) after the station has completed its communications. An important aspect of this approach is flexibility and capability for expansion. A system can be configured with any number of data stations. Also, the IDADS approach allows for interconnecting of subsystems to form a multiple loop system. (See Figure 2.) Each IDADS loop is interconnected by a node terminal. The node terminal performs intercommunication between IDADS loops. In this manner, an integrated system is generated which performs communications between various subsystems.

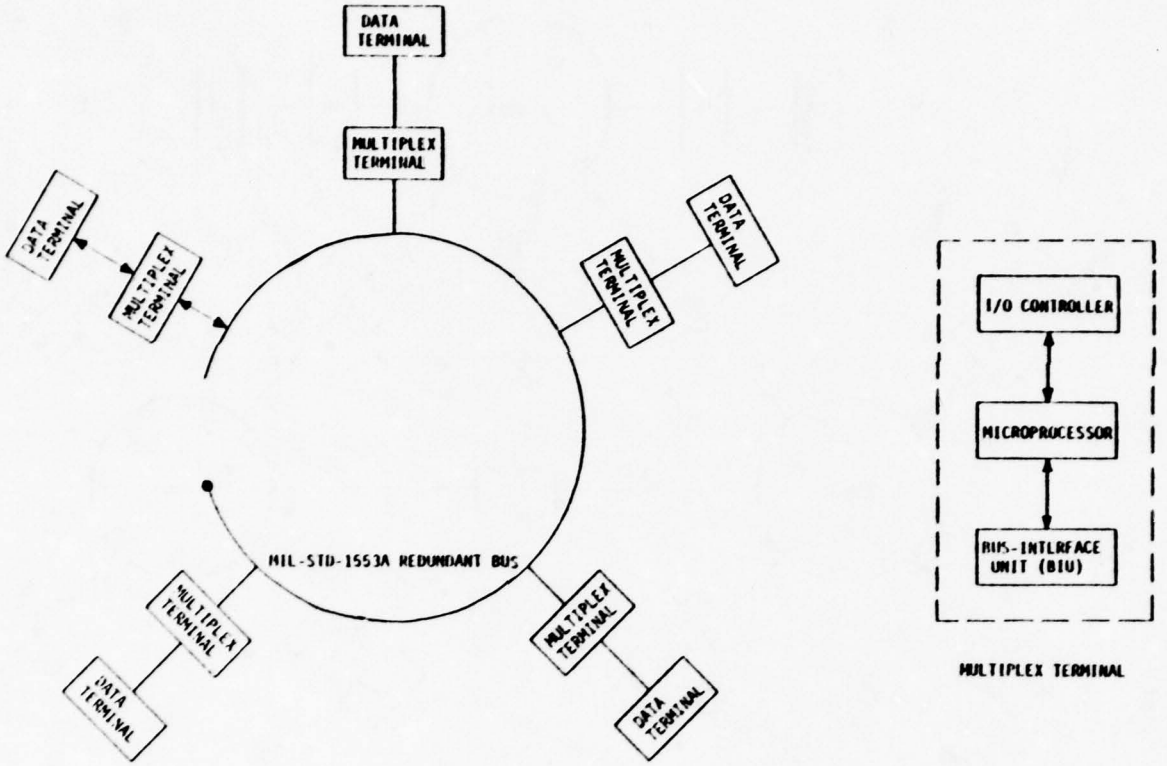



Figure 1. IDADS Configuration

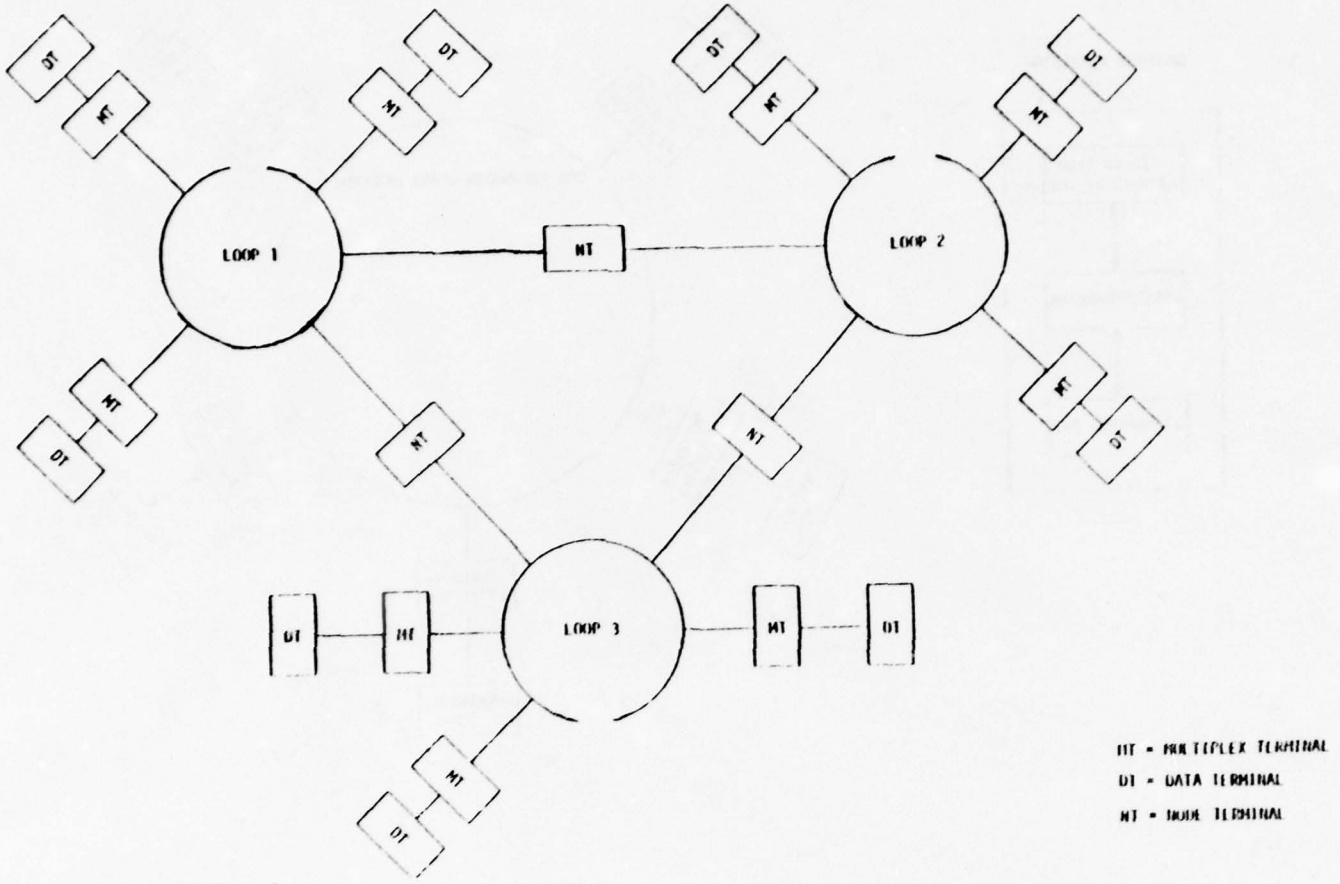



Figure 2. Interconnected IDADS Configuration

The IDADS multiplex terminal was developed to satisfy the requirements of MIL-STD-1553A, MDC 3818, and MIL-G-85013, and is consistent with the proposed 1553B. This was accomplished using a microprogrammable controller. The approach resulted in a design of a multiplex terminal which was used in ULAIDS, F-18 (Stores Management Set and Communication Systems Controller) and the LAMPS MUX BUS TESTER. The F-18 program is a subset of the ULAIDS program such that a ULAIDS terminal could be plugged into an F-18 test fixture and function perfectly. As a matter of fact, multiplex terminals with ULAIDS microcode are used as bus controllers for testing F-18 hardware (SMS and CSC). The ULAIDS program utilizes 432 of the 512 word PROM and the terminal may act as a response terminal or a bus controller, depending upon activation from the mux bus (i.e., receiving a bus offer word) or from its own host CPU (i.e., by being given a forced bus command). The ULAIDS terminal satisfies completely the polled contention requirements of 85013. A terminal which is a bus controller may give up the bus by sending a bus offer word to another terminal. When this terminal receives the word, it interrogates its host computer and either accepts or rejects the bus offer. If it rejects the bus offer, it passes it on to the next terminal in the chain. If it accepts the bus offer, it merely replies with status and assumes command of the bus on which the offer was given. The ULAIDS terminal is also capable of commanding RT to RT transfers.

The LAMPS MUX BUS TESTER contains the same multiplex terminal with a different microcode. Most of the versatile ULAIDS microcode modules have been utilized, however. Here, upon command from the host computer, the terminal acts as either a LAMPS response terminal or as a monitoring terminal. A large variety of options are available in either mode. A bus commander capability has been incorporated into the microcode as a possible future feature for the tester.

## 2.1 Bus Interface Unit

The block diagram of the BIU is as shown in Figure 3. The heart of the multiplex is the controller Z3. The multiplex is connected to two buses, A and B, on

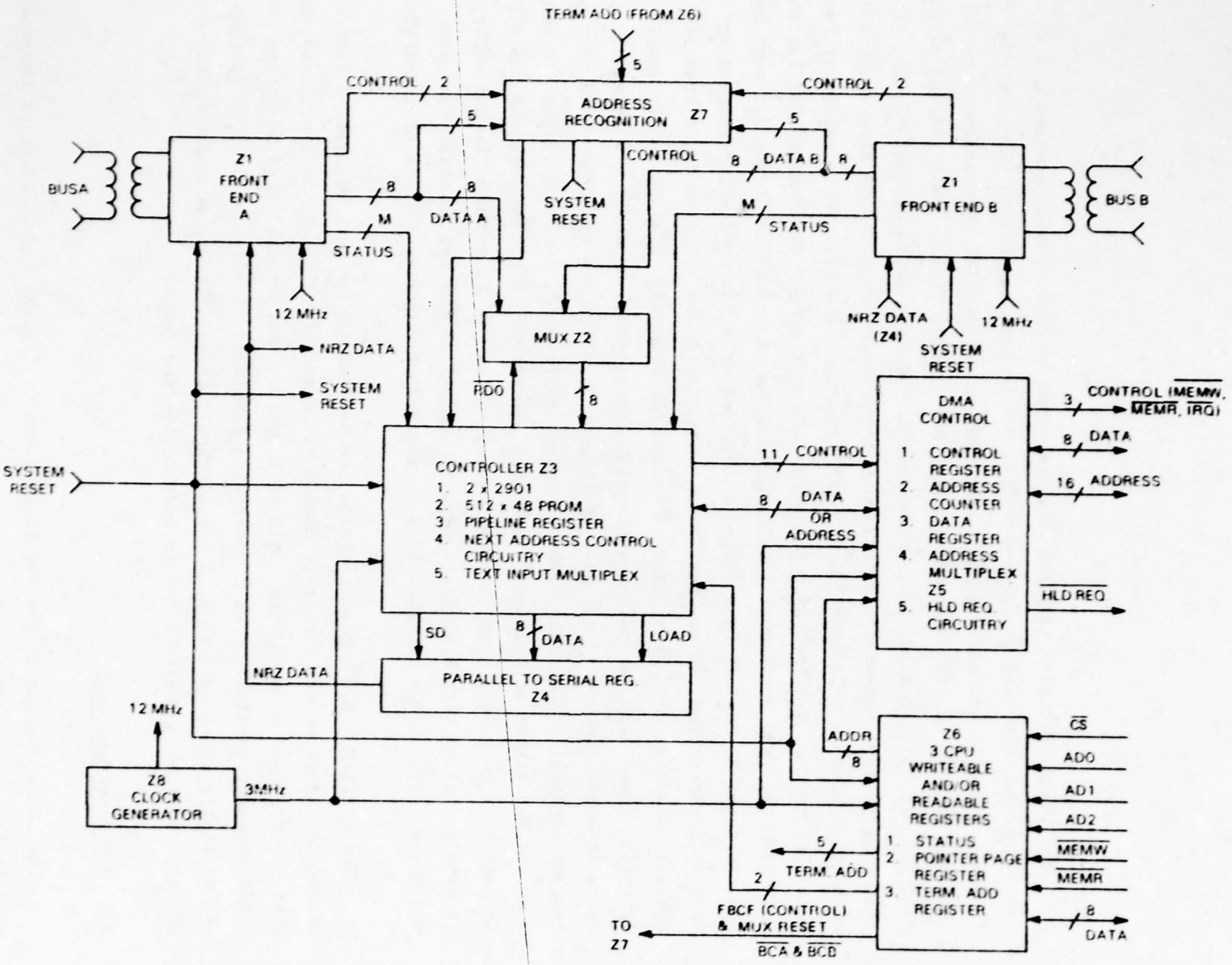



Figure 3 Universal Multiplex - General Block Diagram

one end and has DMA capability to the host computer memory on the other. General signal flow for received messages as a response terminal is through the front end transformer, through Z1, the mux Z2, the controller Z3 and, via the DMA, to main computer memory. For transmitted messages as a response terminal, the flow is from main memory through the DMA, to the controller Z3, into the register Z4 and out via the front end transmitter in Z1. Whether or not a message is for the particular terminal is determined by the address recognition circuitry, Z7. When a valid command sync has been recognized and the correct address is present in the first five bits, a signal is sent to the controller Z3, which forces the ROM address to the RT entry point, 1FF. At 1FF, there is a jump to the RT program and the incoming command word is processed. If the incoming message is a receive message, all the data words are stored indirect and indexed by the word count. If the incoming message is a transmit message, all the data words are fetched indirect and indexed by the word count. In either case, the controller puts out the 8 least significant bits of the block pointer address. The 8 most significant bits are contained in the pointer page register, which is written by the host computer. The 8 least significant bits are formed from the sub-address field of the command word and the T/R bit. Once the pointer is fetched, it is placed in the address counter and the first data byte is placed in (or fetched from) this location. Actually, 2 data bytes are stored (or fetched) on each DMA transfer so that the memory bus is not tied up for long periods of time. The word count is stripped from the command word and stored separately in a 2901 register. When two data bytes are stored (or fetched), this count is decremented. At 0, no more data is handled and the mux prepares to transmit status. Status is fetched from main memory immediately upon address recognition in order to meet 1553A turnaround time (2 to 5 microseconds). The multiplex normally operates in bus interruptable mode as a response terminal. While it is processing one bus, a valid command word on the other bus will cause the mux to switch over and process this data.

The mux, operating as a bus commander, continuously interrogates the main computer to find out if there is a command word ready to be transmitted, and if the terminal is to continue to be a bus controller. When there is a command word ready, the mux places the command word address (lower 8 bits - upper 8 bits from

pointer page register) on the memory bus after gaining control and fetches the command. The terminal determines from the command word the type of transfer required (terminal-to-controller, controller-to-terminal, or terminal-to-terminal) and stores and fetches the data accordingly. When the transfer is complete, the terminal again interrogates the host computer for command word ready (CWR) or command request flag (CRF). If the CRF has been withdrawn, the terminal program jumps to the bus offer routine. Here, the bus offer word is fetched, along with 2 other words. They are the terminal's address and the address of the last terminal in the chain. The bus is then offered to the first terminal in the chain via the bus offer word (subaddress all zeros, word count all zeros). If this terminal responds with status (indicating it has received the message), the mux shuts down after returning to the RT state. If no terminal responds to the bus offer, an error code is set in a word which is written in the host computer memory and an interrupt is issued.

Omitted from the general block diagram is the power-switching circuitry. Since most RT's are active only 5% of the time, considerable power may be saved by switching power onto the proms only when the RT address is recognized. This is done in the case of this terminal.

### 2.1.1 Front End

The front end is shown in Figure 4. It consists of a discrete receiver with a 5-pole filter, a hybrid transmitter, an LSI encoder/decoder, a transformer assembly, and an 8-stage shift register for each channel.

The encoder/decoder, provided by Harris Corp., takes bipolar (Manchester encoded) data from the receiver, validates the sync field, checks for correct Manchester encoding and provides output status signals, NRZ data, and a clock for shifting data. On the transmitting end, it provides a shift clock and a signal which goes true when NRZ data can be accepted [SEND DATA (SD)]. It accepts a transmit command and NRZ data, as well as a level indicating what type of sync is

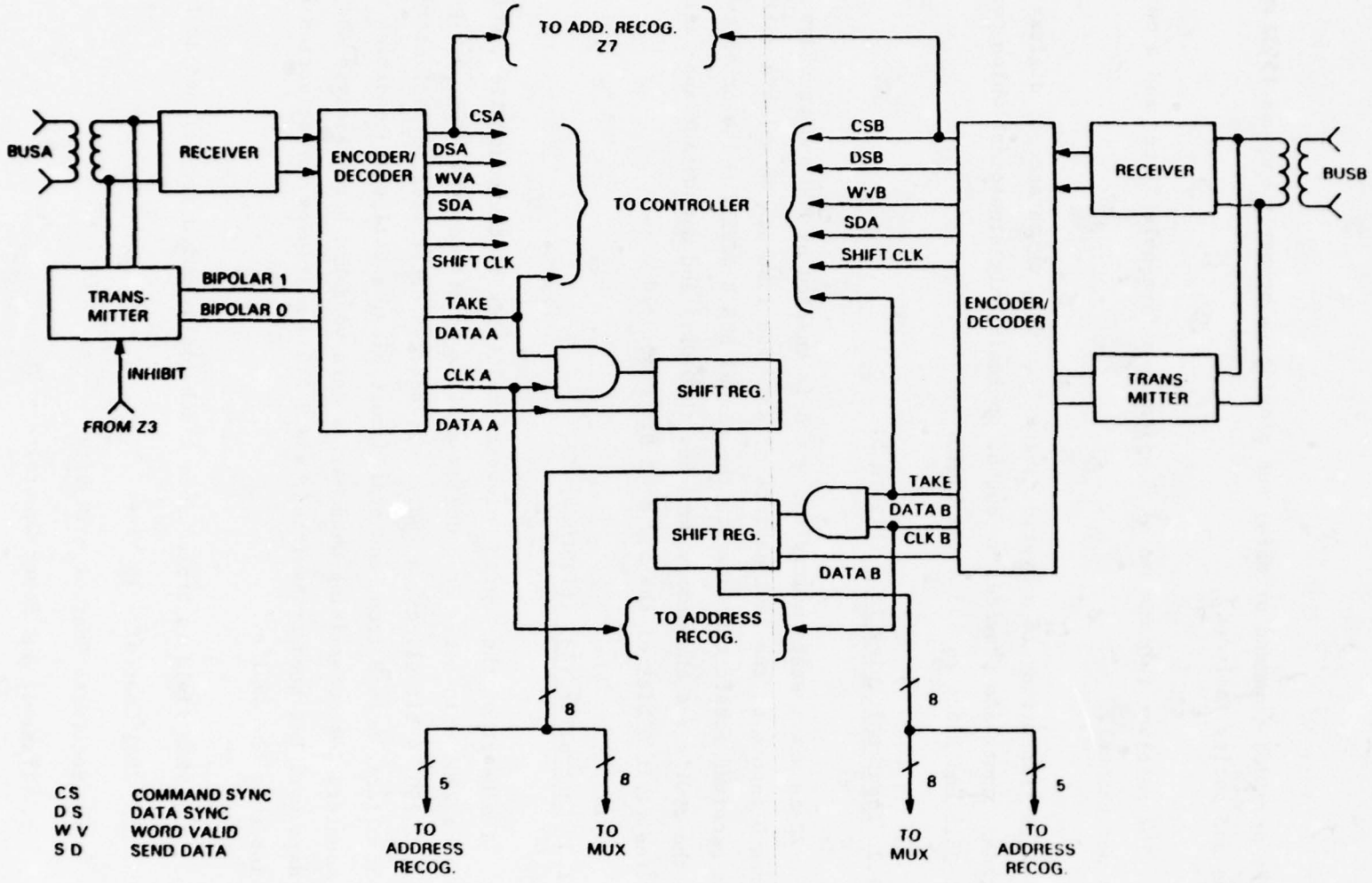



Figure 4. Z1 Front End Block Diagram

to be provided (command or data) and provides a properly encoded 1553A output with sync and parity included.

The receiver is composed of 3 op amps, a five-pole filter and a dual comparator per channel.

The transmitter is a hybrid device from CTI, which accepts digital Manchester encoded inputs and provides an output approximately sinusoidal which conforms to MDC 3818 and hence to 1553A and 85013.

2.1.2 Input Multiplex (Z2) (Figure 5)

Z2 is a tri-state multiplex, which is enabled by $\overline{RDO}$ and provides one of three D inputs to the 2901 devices of Z3. The last bus on which a valid command was received steers the data via the signal BUS B ACTIVE. The other two D inputs to the 2901's are the memory data bus (DB0-DB7) and a constant sub-field, composed of the first 8 bits of the 512 x 48 bit ROM field.

2.1.3 Controller (Z3) (Figure 6)

The heart of the Conrac Universal Multiplex is the controller, which is built up of a 2901 chip set. It contains a 512 x 48 bit control PROM, a pipeline register, 2 2901 4 bit slices, a 29811 next address controller, 3 2911 microsequencer controllers, a 16:8 front-end status multiplex, a 24:1 test condition multiplex, 2 counters for determining when half a data word has been received and when half a data word has been transmitted, and a 2:4 demultiplex for selection of the proper D input to the 2901's.

The PROM field is broken into 4 sub-fields which are defined as follows:

a. 2901 Control - 19 bits

b. Sequencer Control - 18 bits

c. Transmit and Power Control - 4 bits

d. DMA Control - 7 bits

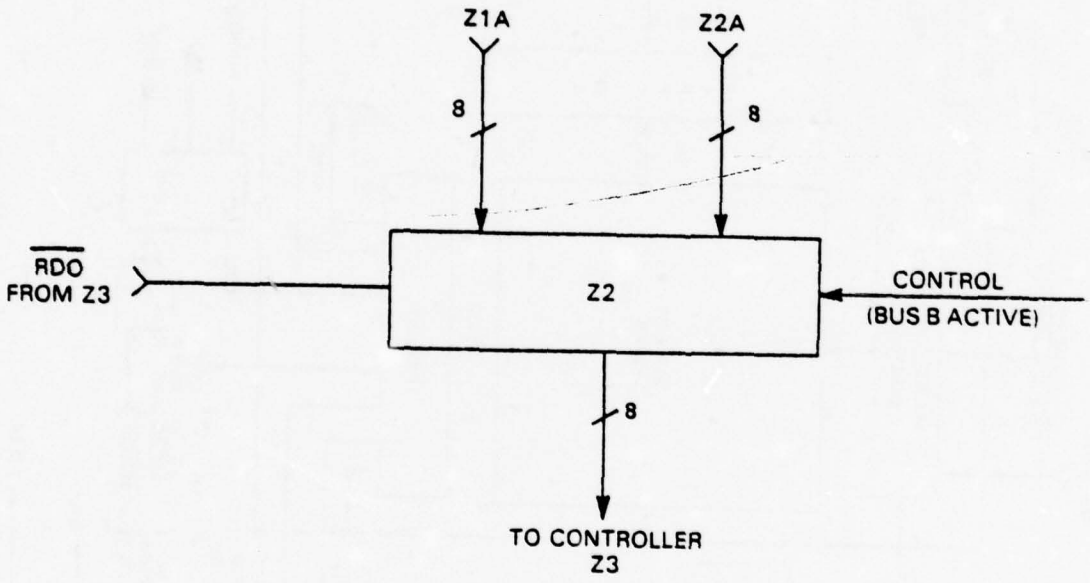



Figure 5 Z2 16:8 Multiplex (Tri-State)

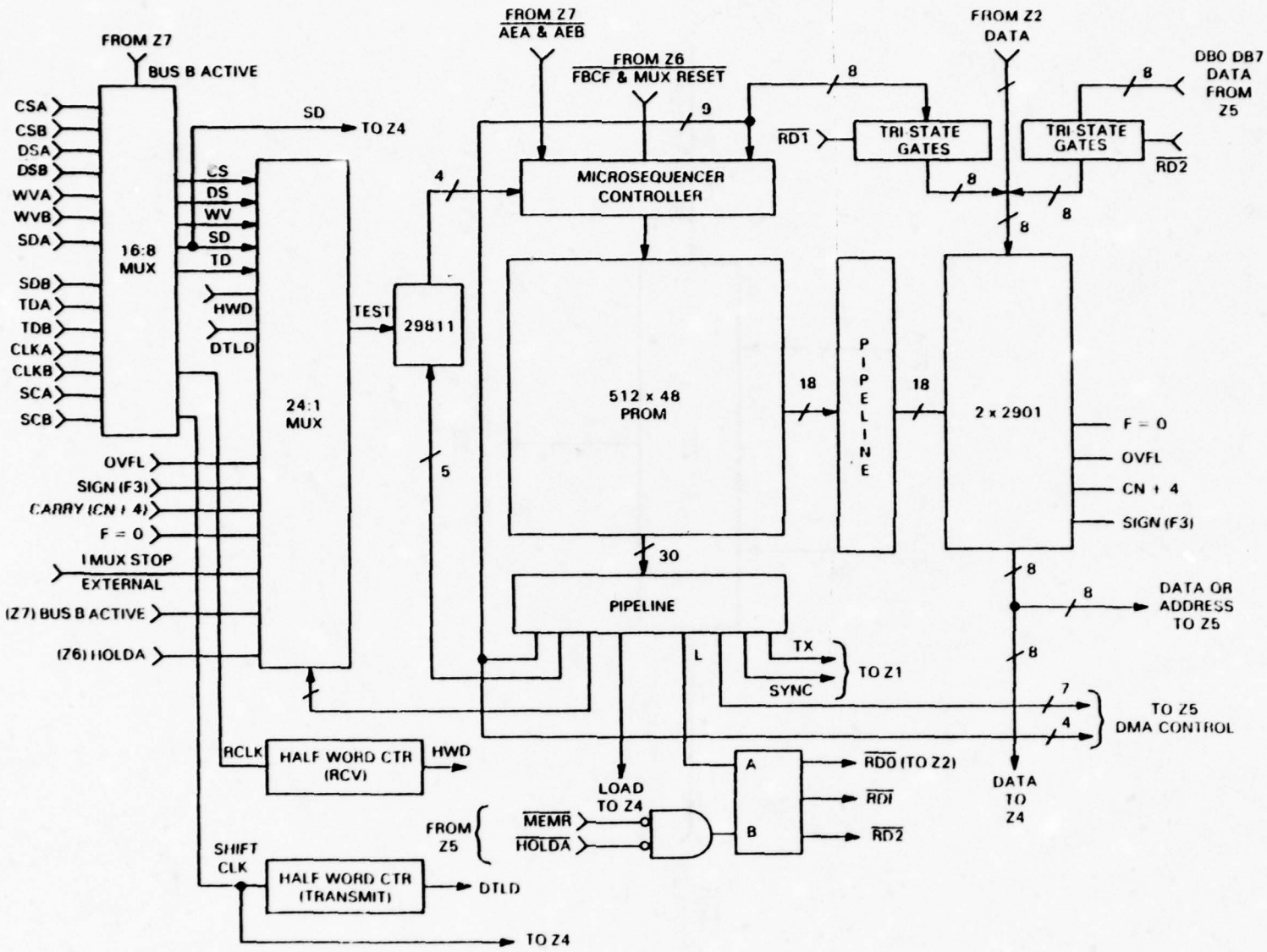



Figure 6 Z3 Controller

Some fields do multiple duty in the program. For example, the next address field (R0-R8) is also used to store constants in the 2901 (R0-R7) and is used as input to the DMA control register (R0-R3).

A list of the instructions which have been defined for the architecture is shown in Table 1. All of the arguments, mnemonics and constants have been defined using the definition phase of AMD's AMDASM meta-assembler. The assembly phase was used next to construct a workable microcode set. Each microword consists of a statement for each of the four fields defined above, separated by "&" (e.g. CLR R0 & CJS CARRY & TRNS & DMWT).

A list of the micro-control field bits is shown in Table 2, along with the description of each bit's function.

The basic instruction clock rate is 3MHZ. Maximum clock rate for the architecture is about 4MHZ.

The controller provides the following outputs to other blocks:

a. 11 lines to control the DMA(Z5).

b. 3 lines to control the loading and shifting of data to the output shift register, Z4.

Three inputs $\overline{AEA}$ + $\overline{AEB}$, FBCF, SYSTEM RESET, and MUX RESET force a jump to locations 1FF, 1FE, 1FD and 1FC respectively. $\overline{AEA}$ OR $\overline{AEB}$ occurs on address recognition when the terminal is an RT and causes a jump to the RT sequence. FBCF is the forced bus command and is a bit written by the host CPU. This forces an entry into the bus commander routine. SYSTEM RESET forces the mux into a reset condition on power turn on, and MUX reset is a bit written by the host CPU which does the same thing.

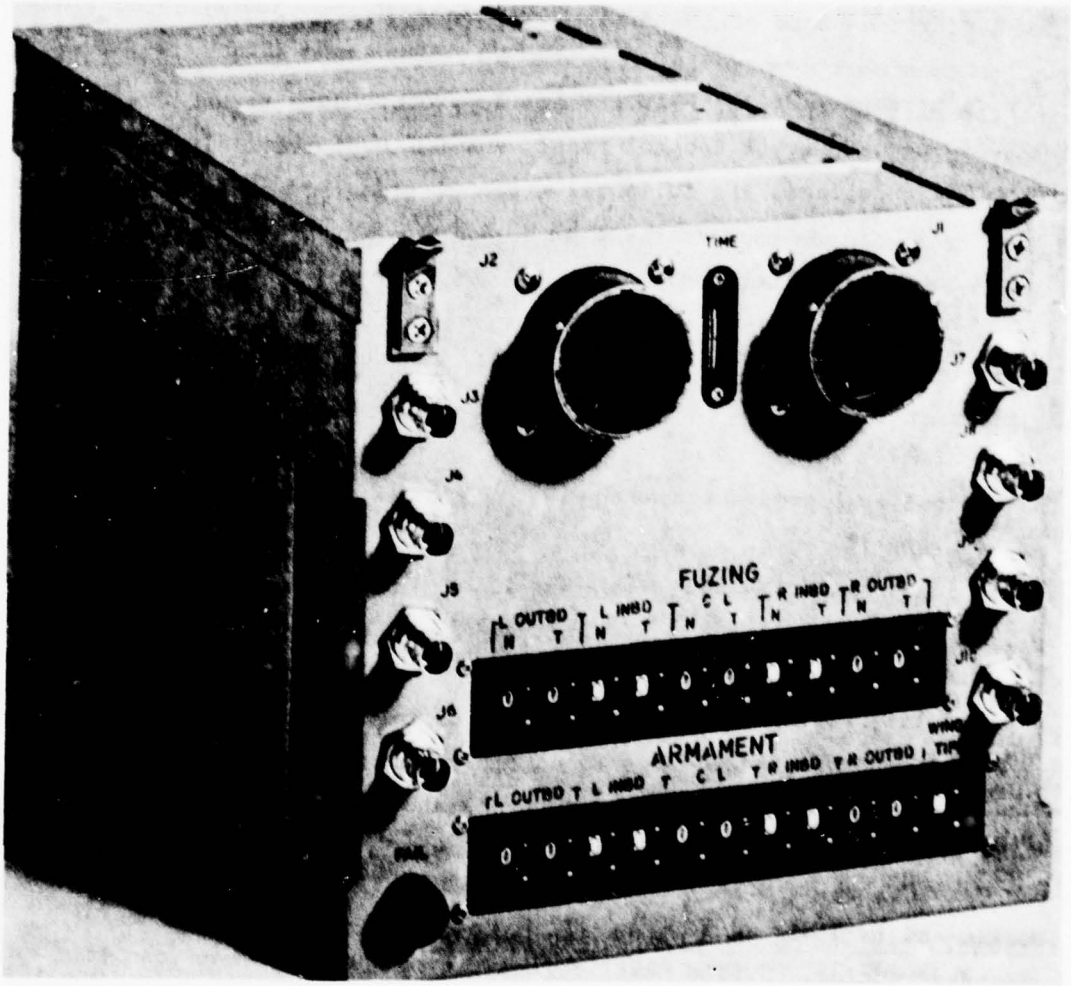



FIGURE 15

ARMAMENT COMPUTER CP-1342/AYQ-9 (V)

Table 1. Arithmatic Instructions (Bits 21 thru 38)

| MNEMONIC | ARGUMENTS | INSTRUCTION |
|---|---|---|
| NOP | AREG, BREG | NO OPERATION Y = AREG |
| CLR | REG | CLEAR REGISTER Y = 0 |
| RDFY | D, REG | D ⟶ REG Y = D; NL, RS, LS, - |
| RDAY | D, AREG, BREG | D ⟶ BREG Y = AREG |
| MOV | AREG, BREG, CNTRL | AREG ⟶ BREG Y = AREA; NL, RS, LS,- |
| ANDD | D, AREG, BREG | 'AND' D ° AREG ⟶ BREG; NL, RS, LS, - |
| INC | REG | Increment REG ⟶ REG |
| DEC | REG | Decrement REG ⟶ REG |
| INCAY | AREG, BREG | Increment BREG ⟶ BREG, Y = AREG |
| DECAY | AREG, BREG | Decrement BREG ⟶ BREG, Y = AREG |
| ROR | REG | Rotate REG right one bit |
| ROL | REG | Rotate REG left one bit |
| OR | AREG, BREG, CNTRL | 'OR' AREG + BREG ⟶ BREG; NL, RS, LS, - |
| AND | AREG, BREG, CNTRL | 'AND' AREG ° BREG ⟶ BREG; NL, RS, LS, - |
| ORD | D, AREG, BREG, CNTRL | 'OR' 'D' + AREG ⟶ BREG; NL, RS, LS, - |
| XORD | D, AREG, BREG, CNTRL | 'XOR' 'D'⊕ARDG ⟶ BREG; |
| XNORD | D, AREG, BREG, CNTRL | 'XNOR' $\overline{D \oplus AREG}$ ⟶ BREG; NL, RS, LS, - |
| XOR | AREG, BREG, CNTRL | 'XOR' AREG⊕BREG ⟶ BREG; NL, RS, LS, - |
| XNOR | AREG, BREG, CNTRL | 'NOR' $\overline{AREG \oplus BREG}$ ⟶ BREG, NL, RS, LS, - |
| DADD | D, AREG, BREG, CNTRL | (D + AREG) ⟶ BREG, NL, RS, LS, - |
| ADD | AREG, BREG, CNTRL | (AREG + BREG) ⟶ BREG; NL, RS, LS, - |
| SUB | AREG, BREG, CNTRL | (AREG - BREG) ⟶ BREG; NL, RS, LS, - |
| | SEQUENCER CONTROL INSTRUCTIONS | |
| JZ | C, CONSTANT | Jump to location zero |
| CJS | C, ADDRESS | Call subroutine if C = T |
| CJP | C, ADDRESS | Branch if C = T |
| PUSH | C, COUNTER VALUE | SAVR - RETURN ADD; If C = T LOAD CNTR |
| JSRP | C, P (ADDRESS) | C = F CALL VIA 'R' REG; C = T CALL VIA 'P' |
| JRP | C, P (ADDRESS) | C = F JMP VIA 'R' REG; C = T JMP VIA 'P' |
| RFCT | C, CONSTANT | C = F NXT ADDR = STACK; C = T NXT ADD = PC; ALU 'D' STACK & CONSTANT |
| RPCT | C, P (ADDRESS) | C = F NXT ADDR = 'P'; C = T NXT ADDR = PC |
| CRTN | C, CONSTANT | C = T STACK = NXT ADDR; CONSTANT |
| CJPP | C, 'P' (ADDRESS) | C = T 'P' NXT ADDR. AND POP STACK |
| LOOP | C, CONSTANT | C = F NXT ADDR = STACK; CONSTANT |
| CONT | C, CONSTANT | CONTINUE; CONSTANT |
| JP | C, 'P' (ADDRESS) | Unconditional jump to 'P' ADDRESS |
| | TRANSMIT/POWER CONTROL INSTRUCTION D, (AR), (BR), CT | |
| TRNS | PWR OFF, XMIT CONTROL | DEFAULT = NOP See TRANS CONTROL BIT DEF |
| | TEST MULTIPLEXER CONTROL INSTRUCTION | |
| TEST | INPUT NAME | DEFAULT = DONT CARR |

Table 1. Arithmatic Instructions (Bits 21 thru 38) (Cont)

DMA CHANNEL CONTROL INSTRUCTIONS

| MNEMONIC | ARGUMENTS | INSTRUCTION |
|---|---|---|
| DMWT | ADDR CNTRL, CONTROL REG LD | (DMA NOP) WAIT |
| DMLD | ADDR CNTRL, CNTRL REG LD | DATA REG LOAD |
| DMRQ | DATA REG LD, CNTRL REG LD | DMA REQUEST |
| DMRL | IRQ, ADDR REG CNTRL, CNTRL REG | DMA RELEASE |

Table 2. Mux Bus Micro-Control Field

| | BIT # | MNEMONIC | PROM & PIN | DESCRIPTION |
|---|---|---|---|---|
| | 0 | R0 | U30 - 9 | |
| | 1 | R1 | \| 10 | A) NEXT ADDRESS FIELD |
| | 2 | R2 | \| 11 | |
| | 3 | R3 | \| 13 | B) LOOP COUNTER LOAD FIELD |
| | 4 | R4 | \| 14 | |
| Q | 5 | R5 PROM6 | \| 15 | C) ALU CONSTANT & MASK FIELD |
| | 6 | R6 | \| 16 | |
| | 7 | R7 | U30 - 17 | |
| | 8 | R8 | U28 - 9 | |
| | 9 | IN0 | U28 - 10 | 29811/2911 |
| | 10 | IN1 | \| 11 | SEQUENCE INSTRUCTION FIELD |
| Q | 11 | IN2 | \| 13 | |
| | 12 | IN3 PROM5 | U28 14 | |
| | 13 | A | U28 - 15 | TEST INPUT MULTIPLEXER |
| | 14 | B | \| 16 | |
| Q | 15 | C | U28 - 17 | ADDRESS FIELD |
| | 16 | S | U24 - 9 | |
| | 17 | T | U24 - 10 | |
| | 18 | TX | U24 - 11 | TRANSMIT ENABLE TO HARRIS DEVICES |
| NS | 19 | LDOUT * | \| 13 | TRANSMIT SHIFT REG PARALLEL LOAD |
| | 20 | SYNC PROM4 | U24 14 | TRANSMIT COMMAND/DATA SYNC SELECT |
| ALU | 21 | I0 | U24 - 15 | ALU SOURCE OPERAND CONTROL |
| | 22 | I1 | \| 16 | FIELD |
| | 23 | I2 | U24 - 17 | |
| | 24 | CN | U26 - 9 | |
| ALU | 25 | I3 | \| 10 | ALU FUNCTION CONTROL |
| | 26 | I4 | \| 11 | FIELD |
| | 27 | I5 PROM3 | U26 - 13 | |
| | 28 | I6 | U26 - 14 | ALU DESTINATION CONTROL |
| ALU | 29 | I7 | \| 15 | FIELD |
| | 30 | I8 | U26 - 16 | |
| | 31 | B0 | U26 - 11 | |
| ALU | 32 | B1 | U19 - 9 | ALU REGISTER SELECT FIELD |
| | 33 | B2 | \| 10 | OUTPUT TO B PORT |
| | 34 | B3 PROM2 | U19 - 11 | |
| | 35 | A0 | U19 - 13 | ALU REGISTER SELECT FIELD |
| ALU | 36 | A1 | \| 14 | OUTPUT TO A PORT |

Table 2. Mux Bus Micro Control Field (Cont)

| | BIT# | MNEMONIC | PROM & PIN | DESCRIPTION |
|---|---|---|---|---|
| | 37 | A2 | U19 - 15 | |
| | 38 | A3 | U19 - 16 | |
| ALU | 39 | L | U19 - 17 | ALU 'D' INPUT MPX LS ADDRESS BIT |
| NA | 40 | CRCL | U9 - 9 | DMA CONTROL REG LOAD ENABLE BIT |
| NA | 41 | IEHRQ | U9 - 10 | DMA CHANNEL REQUEST BIT |
| | 42 | CRF | 11 | DMA CHANNEL RELEASE BIT |
| NA | 43 | DCLK | 13 | DMA CHANNEL DATA REG LOAD ENABLE BIT |
| | 44 | ADDCLK | 14 | DMA CHANNEL ADDRESS REGISTER CLOCK BIT |
| NA | 45 | ACL | 15 | DMA CHANNEL ADDRESS REG PARALLEL LOAD BIT |
| NA | 46 | IRXX | 16 | HOST COMPUTER INTERRUPT REQUEST |
| NA | 47 | ROM PWR | U9 - 17 | ROM POWER OFF BIT |

P1 = U9; P2 = U19; P3 = U26; P4 = U24; P5 = U28; P6 = U30

2.1.4 Parallel to Serial Output Register (Z4) (Figure 7)

This register is loaded via control from Z3 with a particular 2901 output register. Its output is serial and is input to the Harris encoder/decoder. The clock is the encoder shift clock.

2.1.5 DMA Control (Z5) (Figure 8)

The DMA control consists of a memory interface and a CPU interface. The memory interface consists of 16 address lines, 8 data lines and 2 control signals ($\overline{\text{MEMW}}$ and $\overline{\text{MEMR}}$). The CPU interface consists of the hold request ($\overline{\text{HLD REQ}}$), the hold acknowledge (HLDA), the phase 2 clock and the interrupt request.

The DMA is under total control of the controller function Z3 via 11 lines. Lines are provided for loading the control register, loading and clocking the address counter, loading the data register and controlling the CPU interface.

It must be noted that there are 2 address sources, the address counter (16 bits); and the pointer page register (8 bits) and 2901 outputs (8 bits). Which set is enabled to the bus is determined by the control register signal ADDRS.

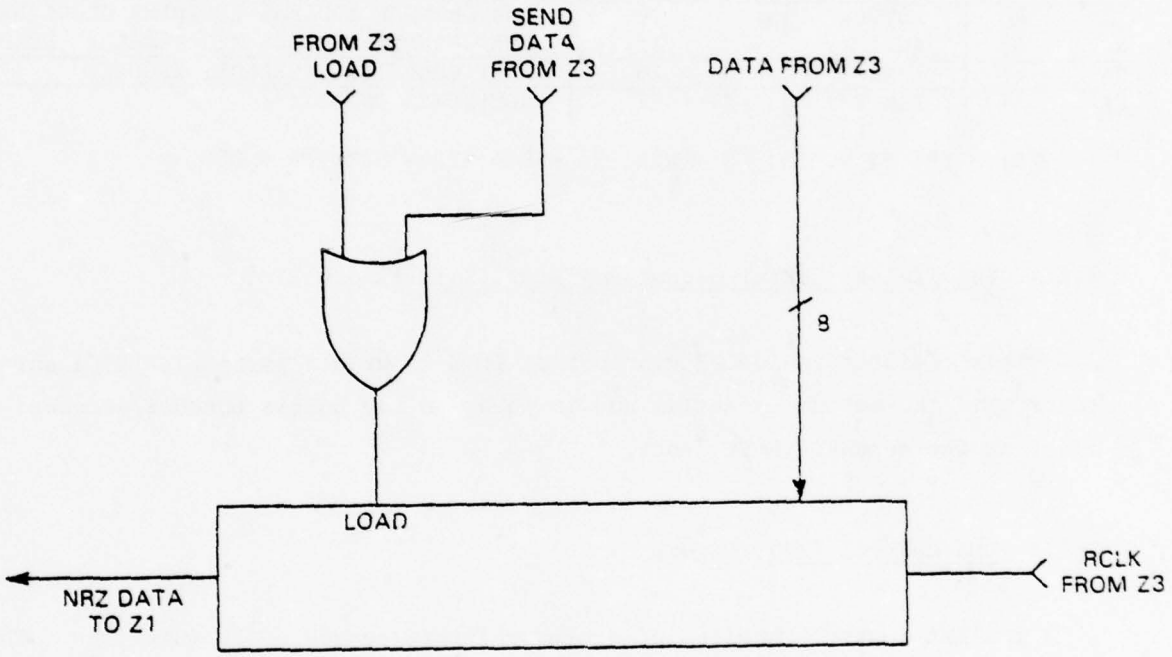



Figure 7 Z4 Parallel to Serial Output Register

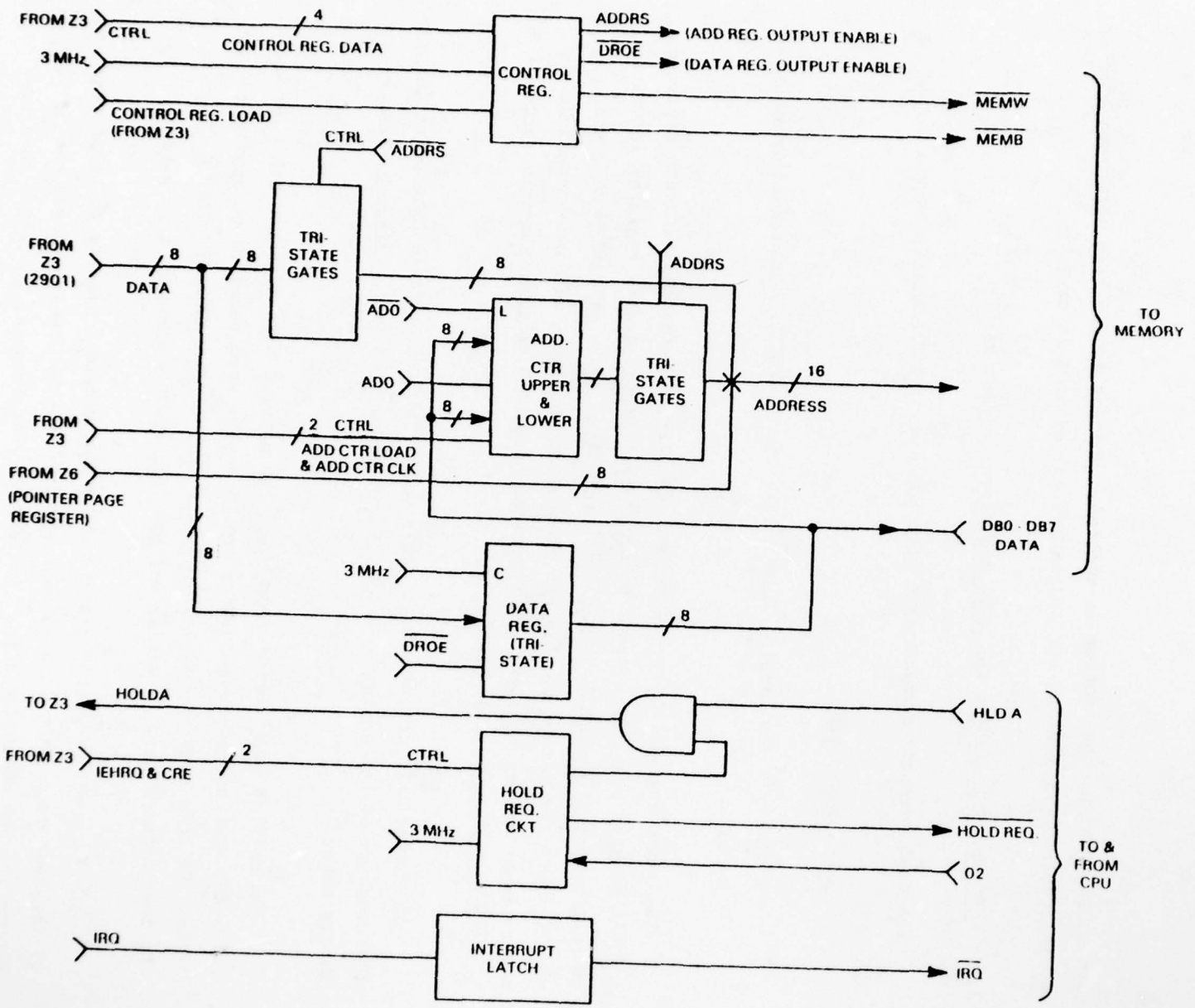



Figure 8 Z5 DMA Control

When ADDRS is high, the counter is enabled. When it is low, the alternate address set is enabled. No address set is enabled unless HLDA is true.

2.1.6 CPU Writable Add/Or Readable Registers (Figure 9)

The host computer communicates with and controls the multiplex via 2 sources:

a. 3 mux registers

b. 2 memory words labeled FLAG WORD 1 and FLAG WORD 2 contained in main memory.

The 3 mux registers are the status register, the terminal address register and the pointer page register. The upper half of the status register is writable but not readable by the CPU. Three control bits may be written - MUX reset, which resets the mux to a quiescent or power down state, $\overline{\text{BCA}}$ which commands the mux to be commander of bus A and forces transactions to take place on this bus and $\overline{\text{BCB}}$ which does the same thing for bus B.

The lower half of the status register contains 2 readable bits which inform the computer of bus activity on A or B since the last read. Reading these bits clears them.

The pointer page register contains the upper 8 bits of the block pointer address for each sub-address. There are 64 double word pointers occupying 128 locations. On this page are also stored FLAG WORD 1, FLAG WORD 2, STATUS (2 words), the last command word (2 words) and the last MODE command word (2 words).

The terminal address register allows the terminal address to be programmable by the host computer. It also contains the FBCF bit, which causes the mux to assume the bus commander role. It is used, for example, in ULAIDS to seize the bus when there has been a lack of bus activity.

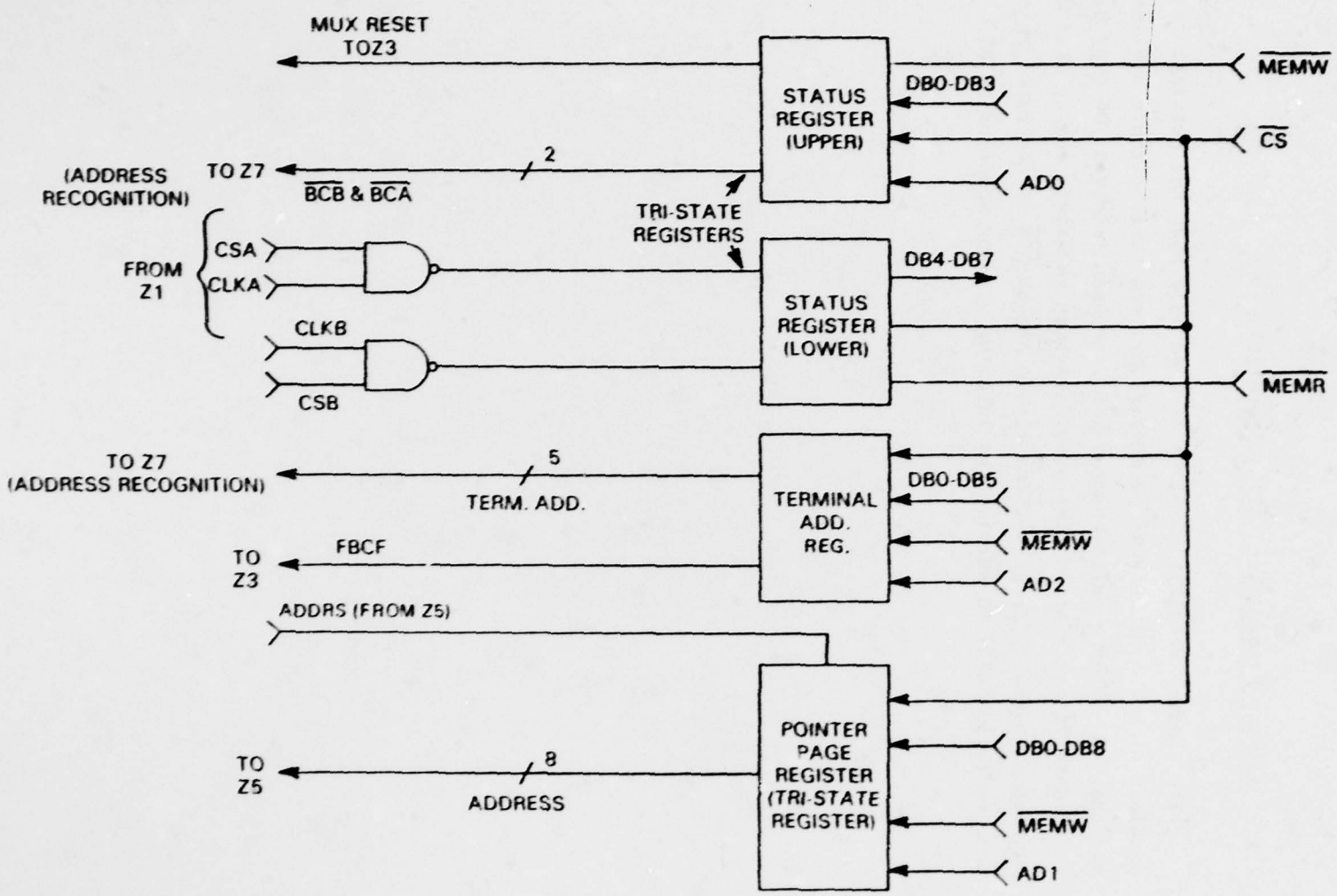



Figure 9 Z6 CPU Writeable &/or Readable Registers

2.1.7 Address Recognition (Z7) (Figure 10)

The address recognition block is used only when the terminal is an RT. It consists of one 5 bit magnitude comparator, one timer circuit and one flip-flop per channel. When a valid command word has been received, CS goes high enabling the counter-timer. After five bits have been received, the counter enables the address comparison. If true, the signal $\overline{AEA}$ or $\overline{AEB}$ is generated which forces the controller (Z3) to service the incoming word or words as an RT.

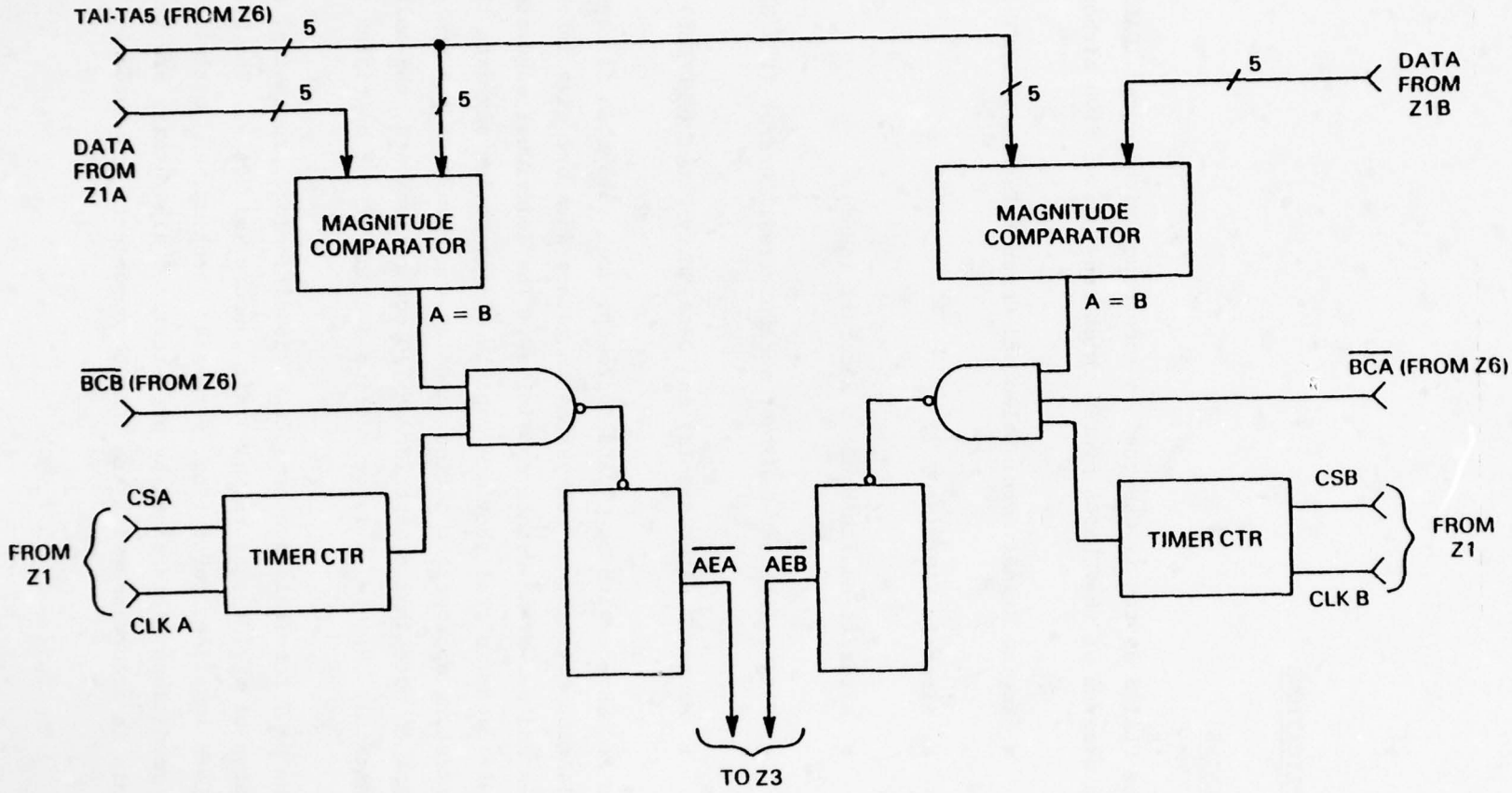



Figure 10 Z7 Address Recognition

## 3. APPLICATIONS

### 3.1 ULAIDS

The ULAIDS system is designed in accordance with Conrac IDADS. Figure 11 is a block diagram of the ULAIDS system which consists of five airborne stations:

- Engine Signal Acquisition and Control Terminal (SACT 1)
- Aircraft SACT (SACT 2)
- Aircraft Health Monitor Recorder (AMHR)
- Flight Incident Recorder and Universal Locator (FIR/UL)
- Master Monitor Display and Data Entry Panel (MMD/DEP).

In accordance with the IDADS approach, each subsystem is capable of initiating communications with other subsystems to gather data and give information regarding failures and out-of-tolerance conditions. The individual subsystems are internally self-sufficient in that they are capable of processing any data that is collected and initiating appropriate action based on the result of the processing. It is important to note that except for an I/O controller card, the multiplex terminals are identical. Figure 12 is an outline drawing of the multiplex terminal.

The SACT terminals are capable of conditioning, processing, and transmitting the status of engine sensors, aircraft sensors and WRA's. The SACT's are programmed to perform sampling, formatting, parameter checking, and evaluation of data. Out-of-limits conditions, determined by predetermined algorithms, are transmitted to other terminals in accordance with the IDAD's communications protocol.

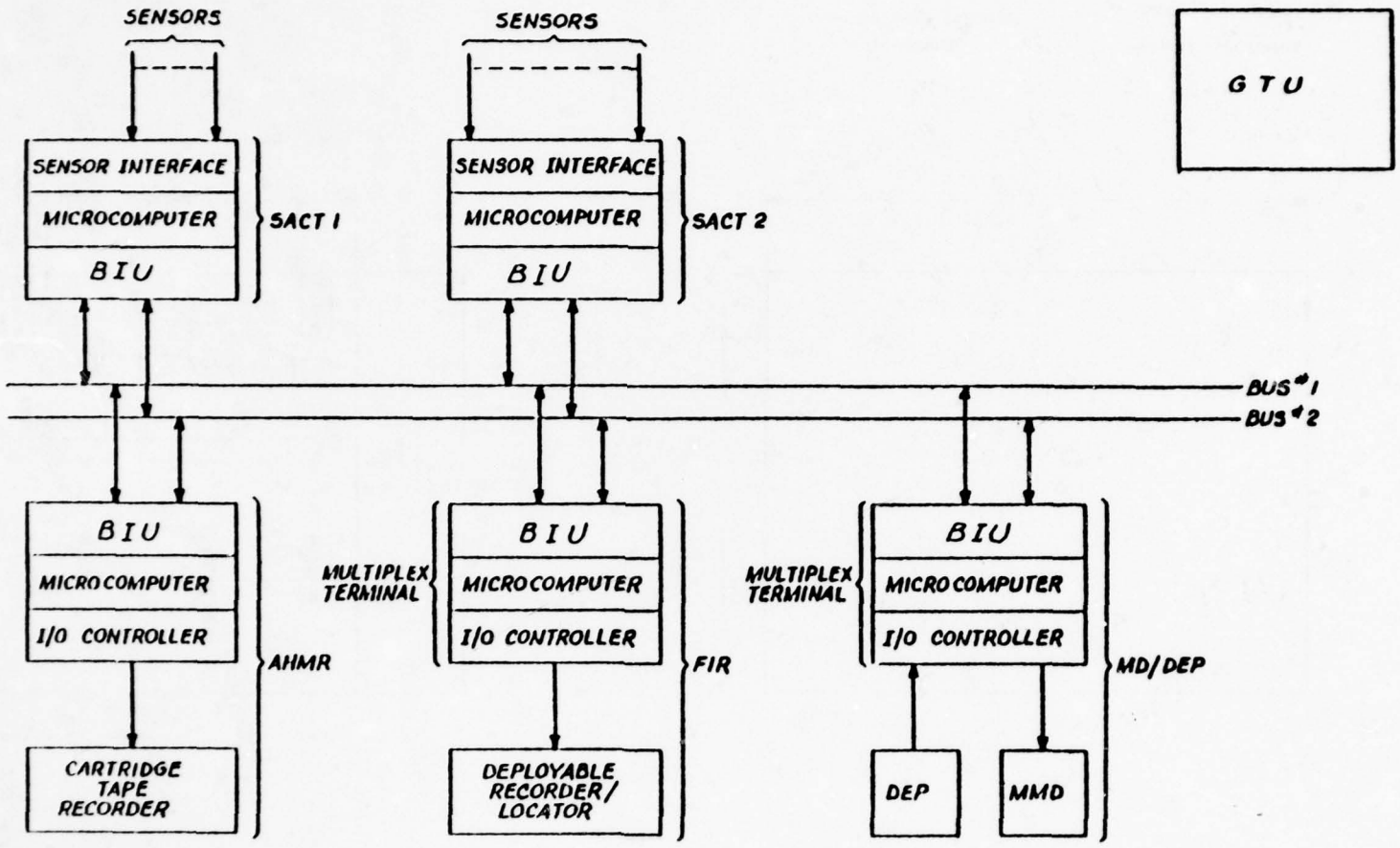
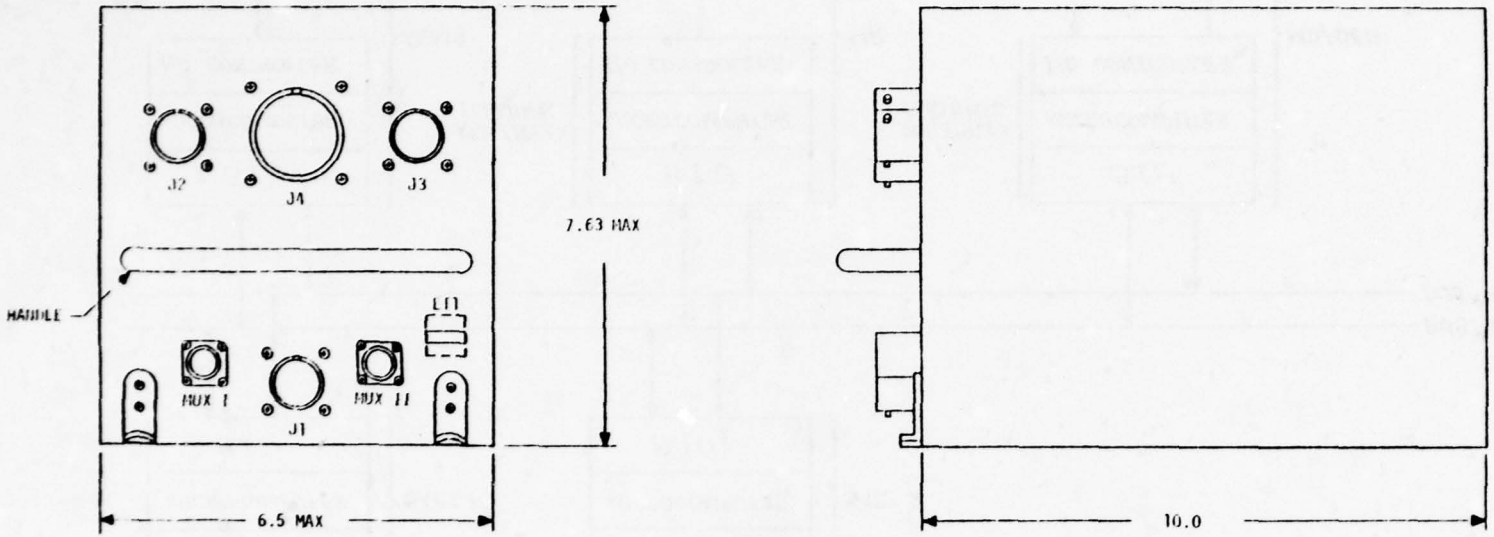



Figure 11. ULAIDS System IDADS Configuration Block Diagram




J1 - POWER
J2 - DEP INTERFACE
J3 - MMD INTERFACE
J4 - VAST

WEIGHT - 15 POUNDS
POWER - 43 WATTS

Figure 12. Multiplex Terminal - MMD/DEP Outline

The MMD/DEP consists of a Data Entry Panel, a Master Monitor Display, and a Multiplex Terminal. Information to be displayed is received by the Multiplex Terminal via the bus and processed in accordance with a prescribed program. The processed data is fed to the symbol generator contained within the multiplex terminal. The output of the symbol generator is a video signal in TV format which is fed to the display unit. Specific data can be called up for display by the Data Entry Panel. The MMD has the capability for generating out-of-limits history information. This information is accumulated from the last initialization and is stored in a non-volatile memory.

The Aircraft Health Monitor Recorder (AHMR) records data which will subsequently be analyzed in a ground terminal. This data will consist of snapshot data and out-of-limits data.

Snapshot data represents a complete 10-second frame of information relating to engine sensors, aircraft sensors, WRA's and equipment status. This data is transmitted at fixed intervals and upon command from the DEP.

Out-of-limits data relates to detection of fault conditions and parameters exceedances as detected by SACT processing. This includes category of the error, location of the error, and dump of the last values of all sensors when the error occurred. The tape recorder has the capabilities of recording 3 hours of data information.

The FIR/UL provides a non-volatile record of audio and data inputs which can be recovered and analyzed in the ground terminal if a crash occurs.

The ULAIDS mission is designed to improve the following operational requirements of the aircraft:

- Flight Safety - by continuous monitoring and display of aircraft subsystems

- Mission Effectiveness - by continuous monitoring and display of current readiness at mission essential systems

- Logistics Efficiency - through the standardization and modularization of equipment and software.

## 3.2 LAMPS Mux Bus Tester

Conrac Model 608T6 Mux Bus Tester is a portable test instrument desinged for use in fault isolation and checkout of aircraft installation using the MIL-STD-1553A command/response or MIL-G-85013 polled terminal multiplex bus configuration. Model 608T6 features a microprocessor controlled diagnostic capability and displays results in alphanumerics on an easily readable front panel. The analyzer (see Figure 13) is contained in a rugged compact suitcase enclosure and features the following capabilities:

- Detect invalid sync fields

- Detect invalid Manchester II codes

- Detect invalid information fields (16 BITS and parity)

- Checking proper response time

- Checking proper protocol between terminals

- Checking BIT error rate as defined in paragraph 4.3.3.2 of MIL-STD-1553A

- Checking proper voltage levels

- Checking proper BIT rate

- Perform diagnostics under software control (microprocessor)

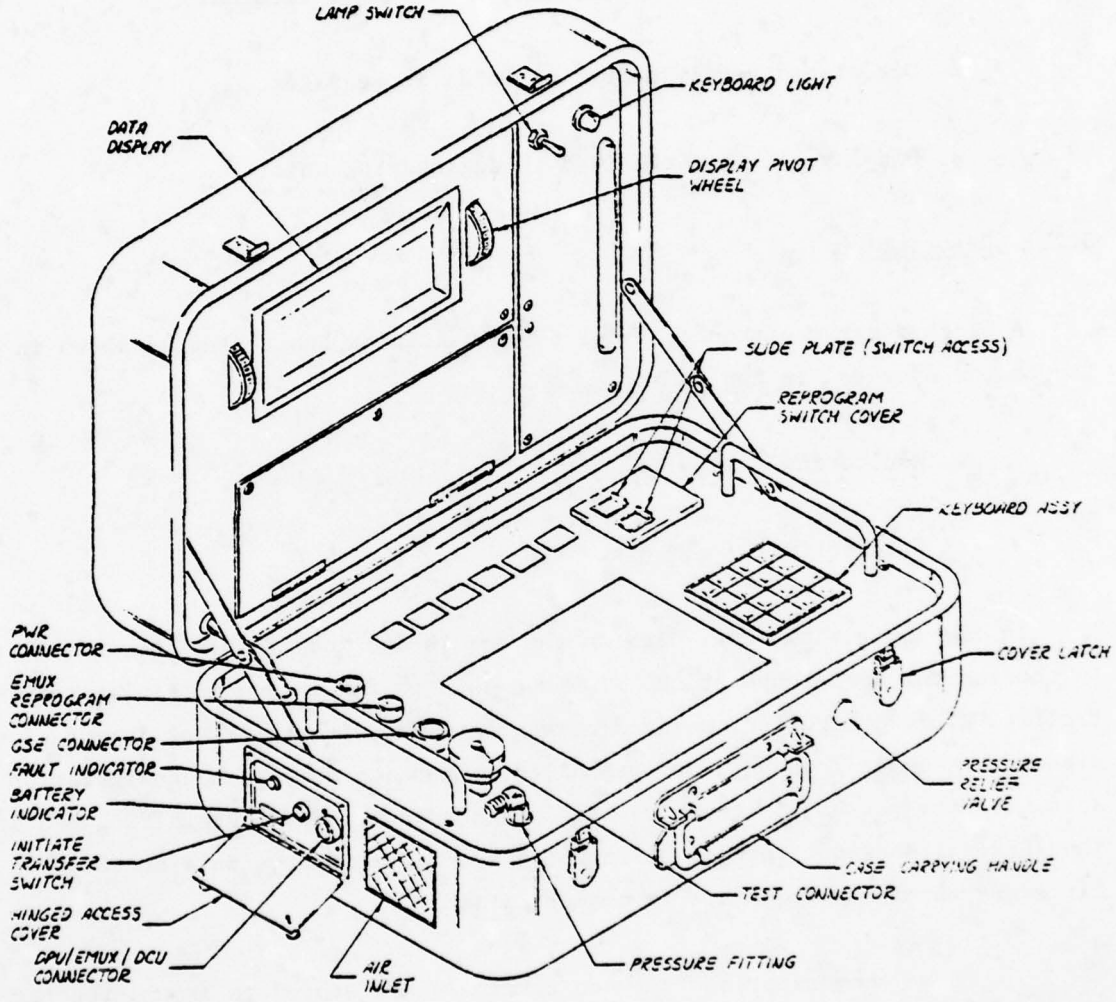



Figure 13. LAMPS Mux Bus Tester

- Display multiplex bus status on an easily readable front panel
- Transmit diagnostic information to remote terminals
- Detect and verify number of words in message
- Display all data received in engineering unit.

### 3.2.1 Description

A block diagram of Conrac Model 608T6 multiplex bus tester is shown in Figure 14. The tester operates in two modes:

- Monitoring
- Interactive.

In the monitoring mode, data on the bus is fed into the tester and all traffic is checked for normal operation. The bus data is fed into a multiplex terminal controller. This controller, using a microprogrammer, is capable of interfacing with either the command/response or the polled contention multiplex configuration and reacts with the bus as if it were a standard terminal. The basic difference is that the terminal does not respond to requests but rather stores data received or transmitted by terminals which are in communication.

These communications between terminals are stored in the microporcessor core memory and can be called up for display using the data entry keys contained in the unit. The block diagram shows that the data from the multiplex bus is also fed to the signal analyzer which monitors the line condition and detects the following conditions:

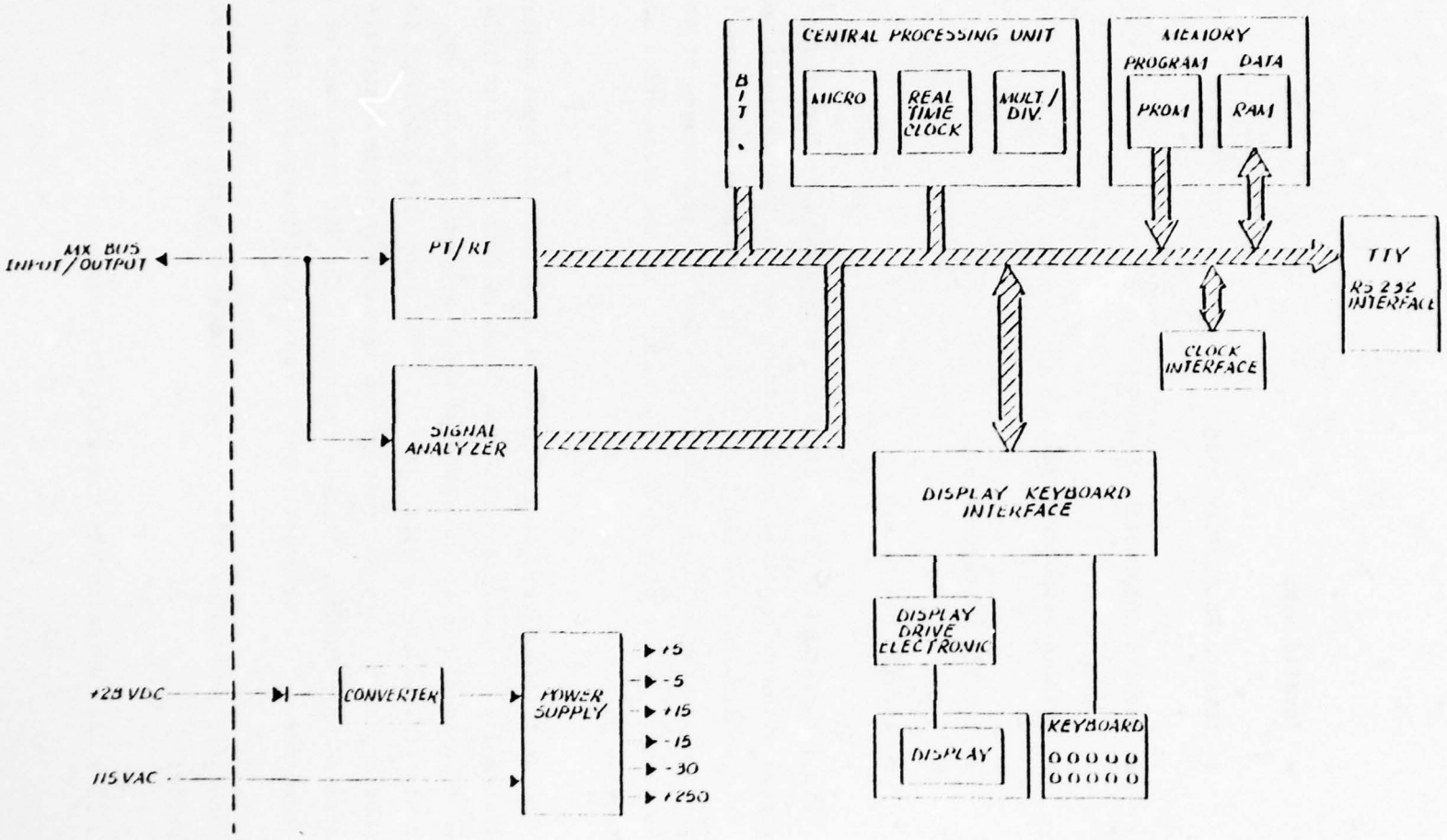



Figure 14. LAMPS Multiplex Bus Tester

- Invalid sync
- Invalid Manchester code
- Invalid information field
- Invalid response time
- Invalid voltage levels
- Invalid BIT rate.

This information is also fed to the microprocessor for storage in memory and correlated with the specific communication between the terminals under consideration. This is also displayed on the front panel in an alphanumeric format. The system can also be programmed to perform diagnostics on the messages if the specific content of the message is predetermined. The results of the diagnostics are presented on the display panel.

In the interactive mode the tester acts as an additional multiplex terminal in the system and transmits and receives messages from other terminals. Varied diagnostic techniques can be incorporated under program control. One approach allows the tester to request specific data from the terminals and check for proper content. A second approach is to transmit canned messages to other terminals to be used for retransmission. In both of these techniques the BIT error rate of the system can be determined. The diagnostic results are displayed on the front panel.

Specific test modes that can be accomodated in the interactive mode are the following:

- Add noise to the transmitted message

- Generate an invalid sync in the transmitted message
- Generate invalid Manchester code in the transmitted message
- Generate invalid message field in the transmitted message
- Generate various voltage levels in the transmitted message
- Generate invalid BIT rate.

## 3.3 F-18 Communications System Control Set C-10382/A

The Communication System Control Set (CSC) being designed by Conrac for the USN/MCAIR F-18 is the total interface between the AN/AYK-14(V) Mission Computer (MC), the pilot operated Up Front Control (UFC) panel and the 17 associated Communication, Navigation and Identification (CNI) subsystems on board the aircraft. The CSC provides operator and computer control of these systems and the transfer of data.

Operations in the CSC involve conversion, storage and transfer of digital, analog, synchro and discrete data.

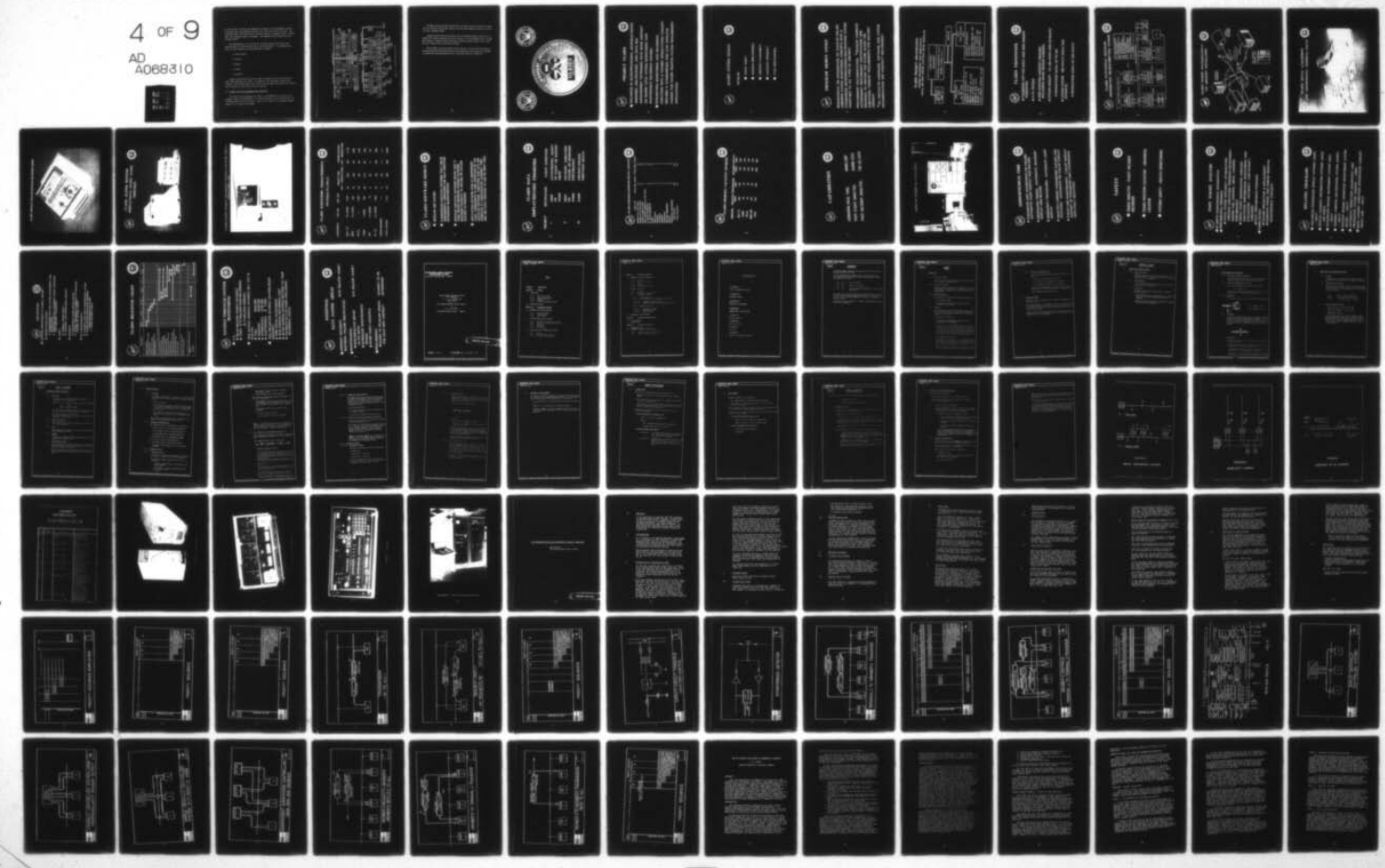
The F-18 CSC provides the intelligence and power to the UFC panel and in terms of energy management powers up/down those CNI subsystems as required by the pilot through the UFC panel.

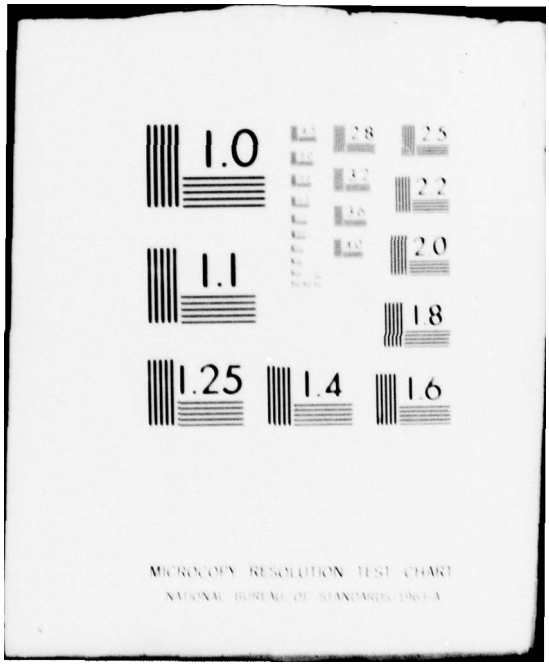
The concept of the CSC is to integrate control of all CNI equipments, thereby eliminating proliferation of cockpit controls, and consolidating these controls into the UFC panel for reduced pilot workload.

The doubly-redundant MIL-STD-1553A multiplex (MUX) connections from the F-18 CSC to the AN/AYK-14 Mission Computer (MC) provides a port for flow of information and control between the MC and the CSC and CNI equipment. Flow of control can only

AD-A068 310 AERONAUTICAL SYSTEMS DIV WRIGHT-PATTERSON AFB OH F/G 17/2
PROCEEDINGS OF THE AFSC MULTIPLEX DATA BUS CONFERENCE (2ND), HE--E
OCT 78 E C GANGL
UNCLASSIFIED ASD-TR-78-34-VOL-2 NL

4 OF 9
AD A068310




1.0
1.1
1.25
1.4
1.6
2.8
2.5
2.2
2.0
1.8
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

be from the MC to the CSC, but information of display on the alphanumeric display of the UFC panel (scratchpad and options) also flows from the MC to the CSC. The information that flows from the CSC to the MC consists of equipment status, received data, and operating options of equipment. BIT commands and BIT status also flow through the MUX.

The requirements of the F-18 CSC are to process data to and from the CSC, Mission Computer, Up Front Control Panel, and CNI equipment. To accomplish this task, the CSC interfaces the following types of signals:

- Serial Digital
- Discrete
- Analog
- Synchro
- Multiplex.

Besides interfacing these various types of signals the F-18 CSC processes at least 1300 parameters per second. To accomplish this requirement, Conrac's CSC contains a fast microprocessor which controls and processes the required data leaving over 80 percent of real time available for growth.

## 3.4 US Navy F-18 Stores Management Set AN/AYQ-9(V)

The Conrac Stores Management Set (SMS) for the USN/MCAIR F-18 consists of an Armament Computer CP-1342/AYQ-9(V), two Wing Tip Decoders, four Pylon Decoders, two Fuselage Decoders and a Gun Decoder. Figure 15 is a photo of the Armament Computer. Figure 16 is a system block diagram.

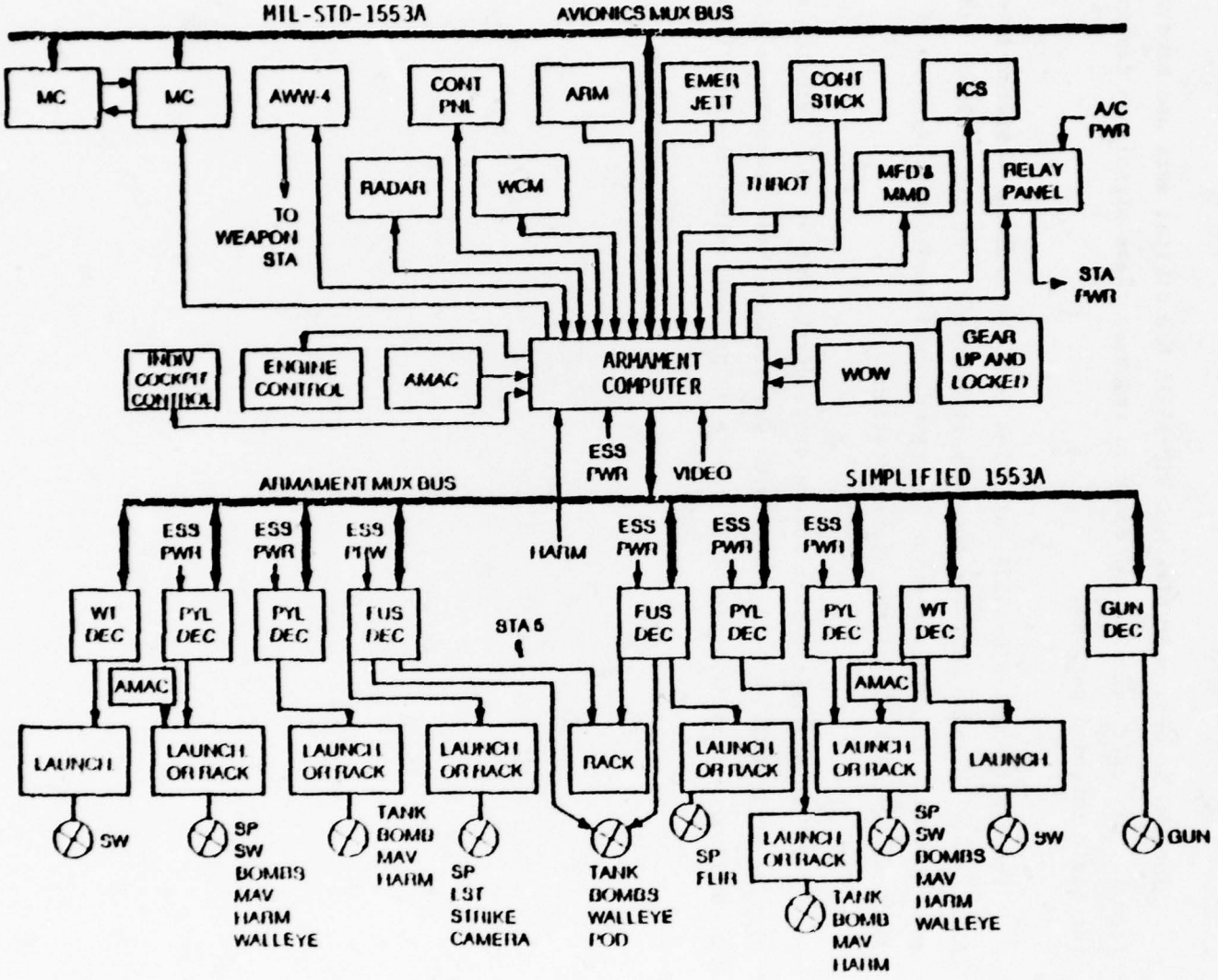
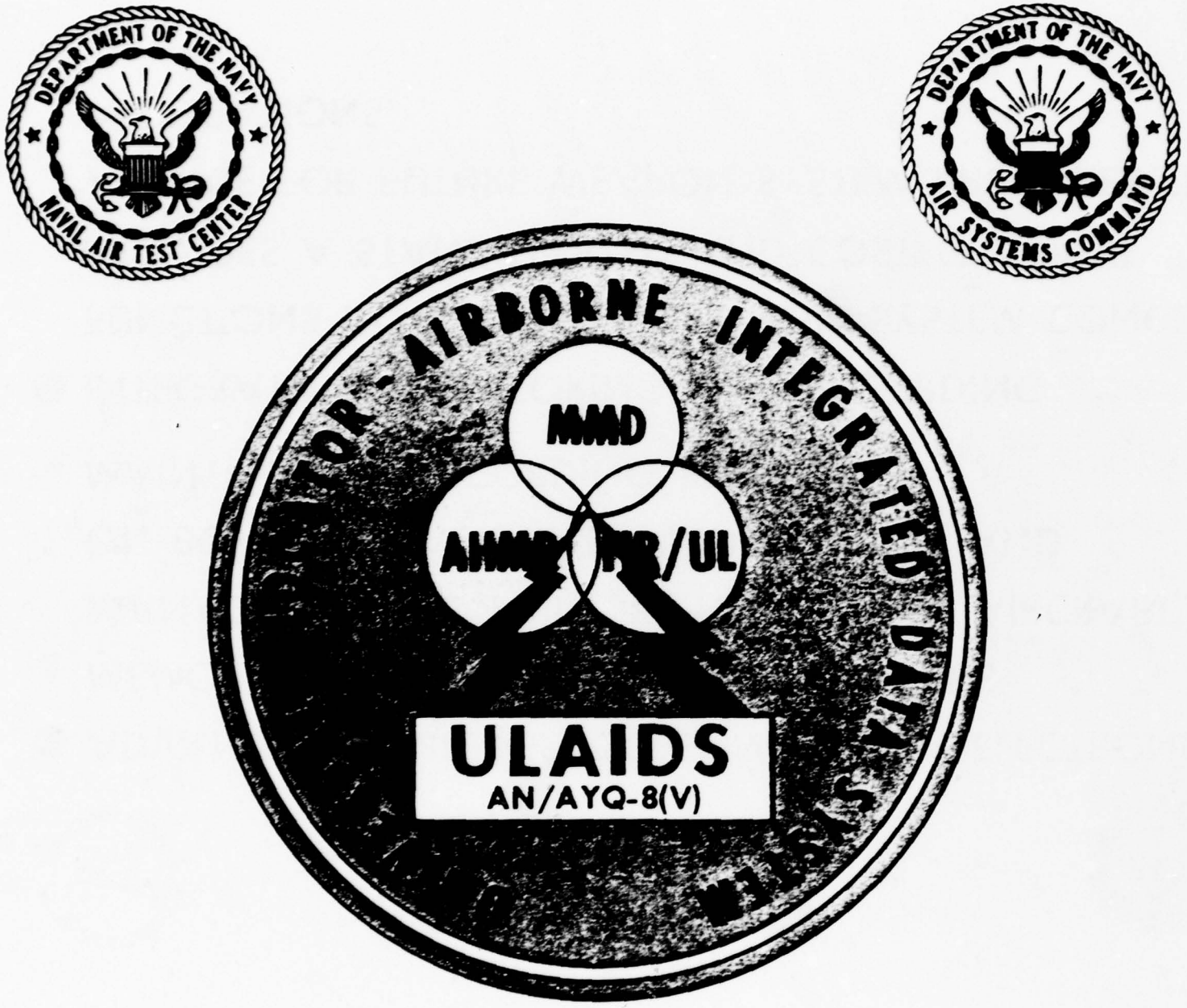



Figure 16. F-18 SMS Interface Diagram

The SMS accepts multiplex MIL-STD-1553A bus digital data and hard wire signals from the aircraft avionics and controls and uses these signals to deploy the stores for their intended purpose.

Through the MIL-STD-1553A multiplex bus, the Armament Computer interfaces the AN/AYK-14(V) Mission Computer and Cockpit Controls. On the outboard side, the Armament Computer, through a special low power, dedicated Conrac version of the 1553A Mux Bus, controls all of the Stores Stations.

The F-18 SMS is the most modern state-of-the-art of such avionics systems employing digital MIL-STD-1553 multiplex and microprocessor techniques with designed in Reliability and Maintainability and a 98 percent self-test capability.


DEPARTMENT OF THE NAVY
NAVAL AIR TEST CENTER
DEPARTMENT OF THE NAVY
AIR SYSTEMS COMMAND
MMD
/UL
ULAIDS
AN/AYQ-8(V)

# PROJECT ULAIDS

- ADVANCES IN MICROPROCESSING, MICROELECTRONIC MEMORIES, MULTIPLEXING, AND DATA BUS MANAGEMENT TECHNIQUES HAVE MADE AIRCRAFT ON BOARD TOTAL SYSTEM MONITORING AND MAINTENANCE RECORDING FEASIBLE.
- INTEGRATING MONITORING AND RECORDING FUNCTIONS INTO A FLEXIBLE SINGLE-SYSTEM CONCEPT PROVIDES A STANDARDIZED AND COST EFFECTIVE AVENUE FOR FUTURE WEAPON SYSTEM PROCUREMENT APPLICATIONS.




# ULAIDS SYSTEM GOALS



INCREASE

- FLIGHT SAFETY
- MISSION EFFECTIVENESS
- AIRCRAFT AVAILABILITY
- LOGISTICS EFFICIENCY
- COST EFFECTIVENESS

# PROGRAM DESIGN GOALS

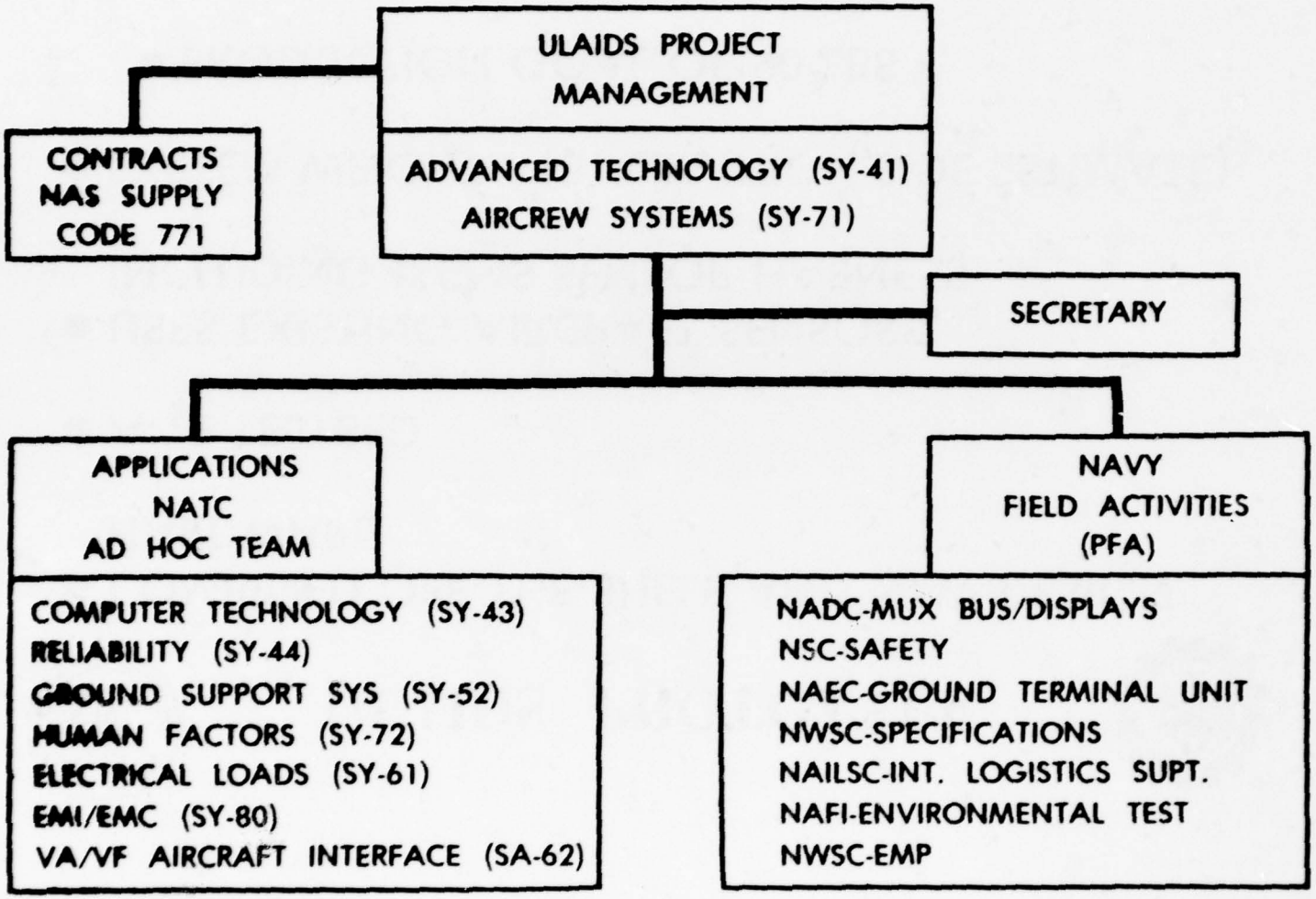
DEMONSTRATE THE INTEGRATION OF STATE-OF-THE-ART AIRBORNE SYSTEM MONITORING CONCEPTS INTO A STANDARIZED MICROPROCESSING/MULTIPLEXING SYSTEM ADAPTABLE TO ALL TYPES OF AIRCRAFT.

PROCUREMENT DOCUMENTATION: ★DEVELOP COMPLETE GOVERNMENT PROCUREMENT SPECIFICATIONS AND DRAWINGS TO PROVIDE FOR FUTURE ULAIDS (OR SUBSYSTEMS THEREOF) PROCUREMENTS FOR MILITARY AIRCRAFT

★ALL PROTOTYPE HARDWARE, SOFTWARE AND SYSTEM DOCUMENTATION ARE CONTRACT DELIVERABLES.

# ULAIDS PROJECT ORGANIZATION

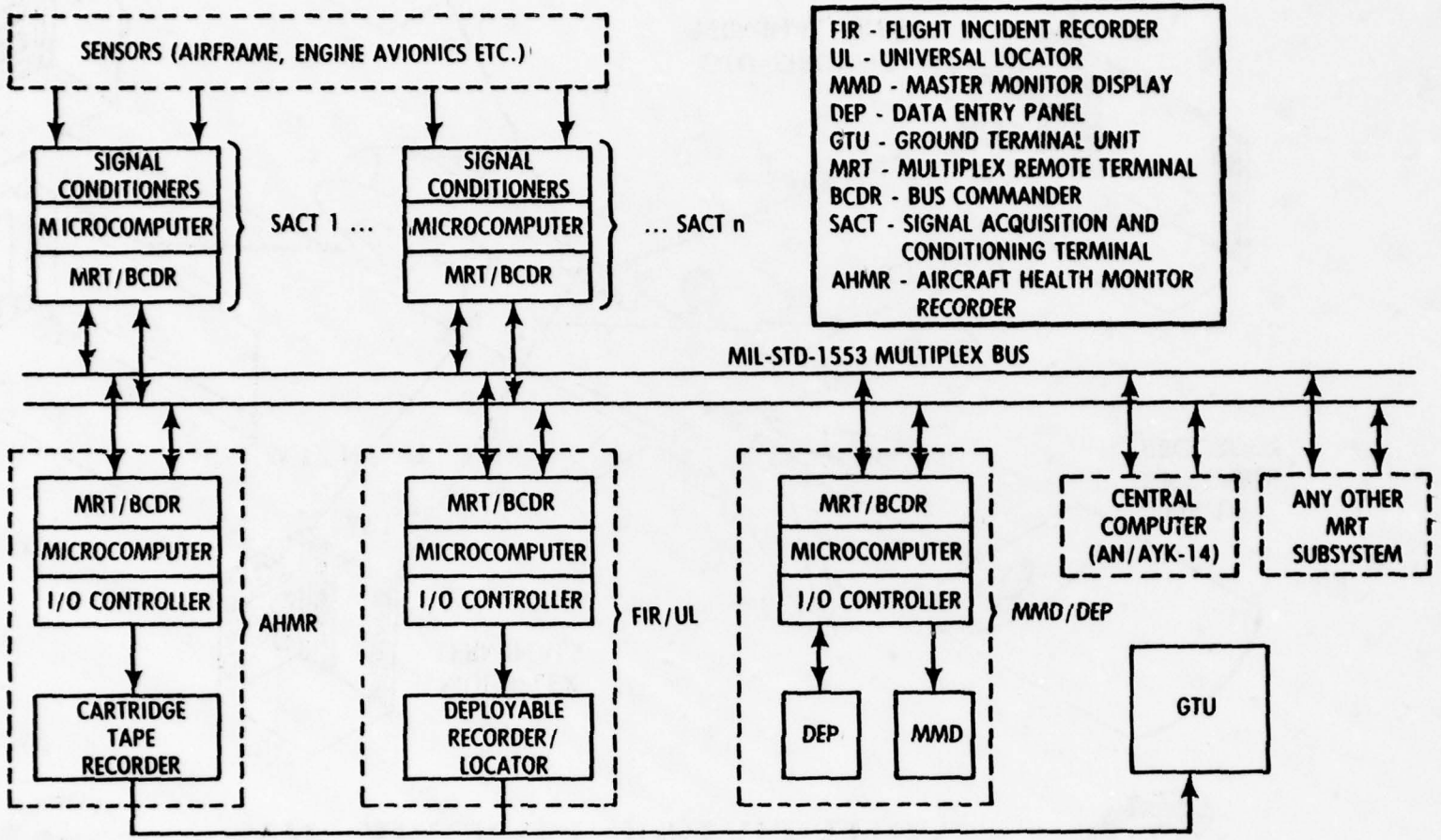
## SYSTEMS ENGINEERING TEST DIRECTORATE
## NAVAL AIR TEST CENTER, PATUXENT RIVER, MARYLAND



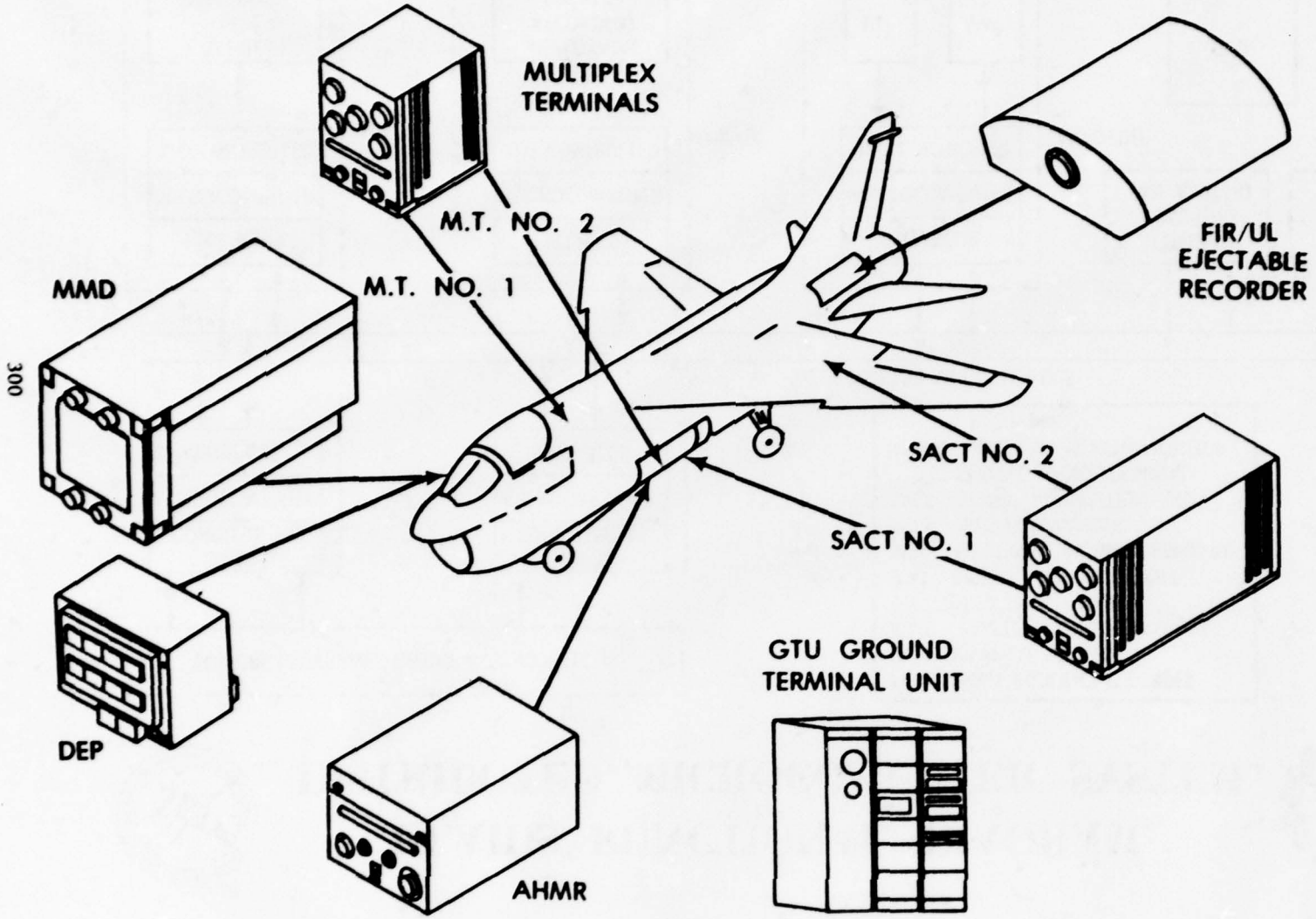
# ULAIDS PROTOTYPE

- COMBINED OFF-THE-SHELF AND NEW DESIGN HARDWARE
- A-7E TESTBED
- USES EXISTING AIRCRAFT SENSORS, INCLUDING IECMS SENSOR HARNESS
- SYSTEM WEIGHT: 131 LBS (HIGH SIDE ESTIMATE)
  - PRODUCTION GOAL OF 90 LBS
- SYSTEM POWER: 488 WATTS (HIGH SIDE ESTIMATE)
  - PRODUCTION GOAL OF 330 WATTS

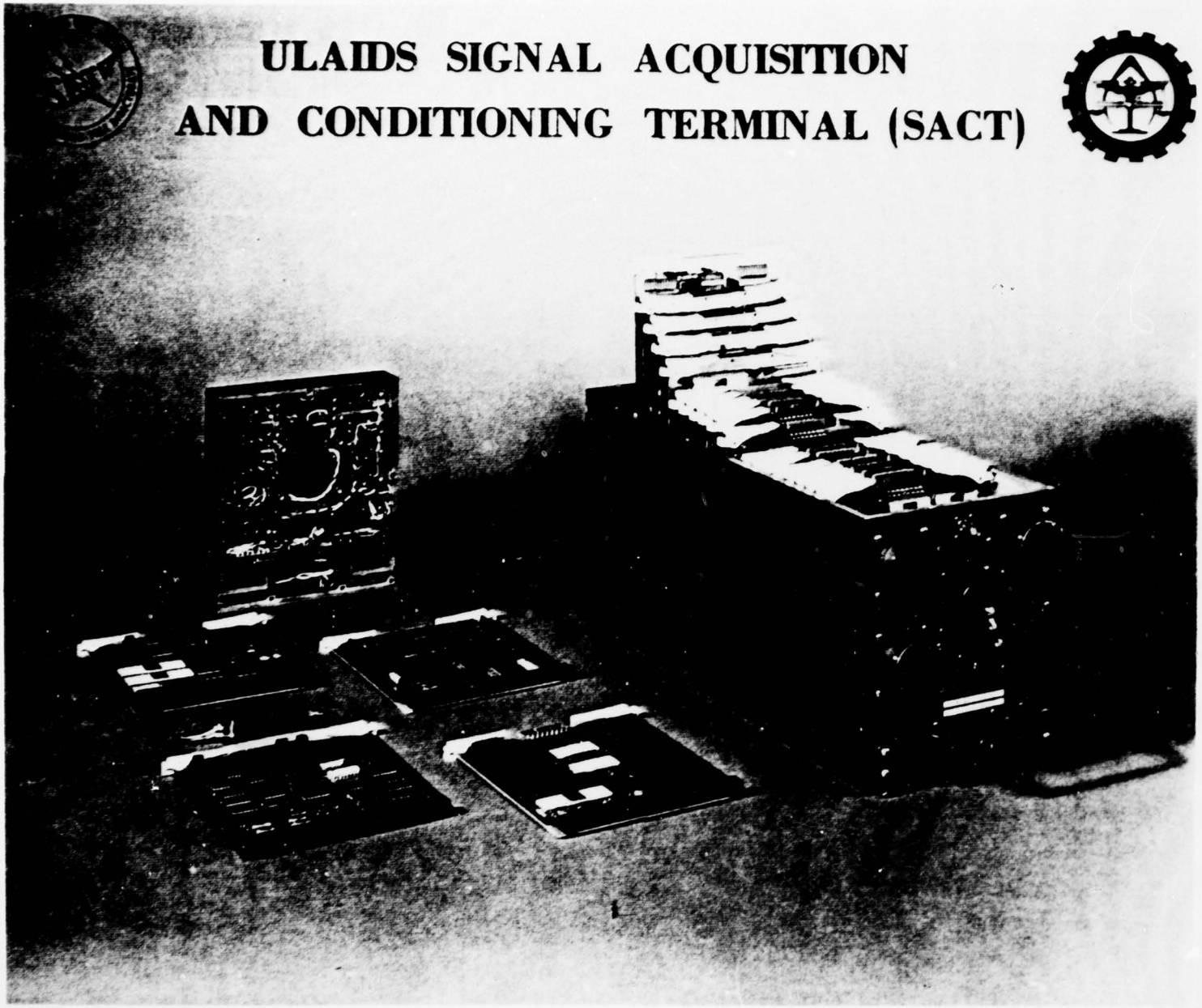
# ULAIDS FUNCTIONAL DIAGRAM, DISTRIBUTED MICROCOMPUTER SYSTEM



# ULAIDS SYSTEM AN/AYQ-8(V)(XN-1)
# A7E AIRCRAFT INSTALLATION



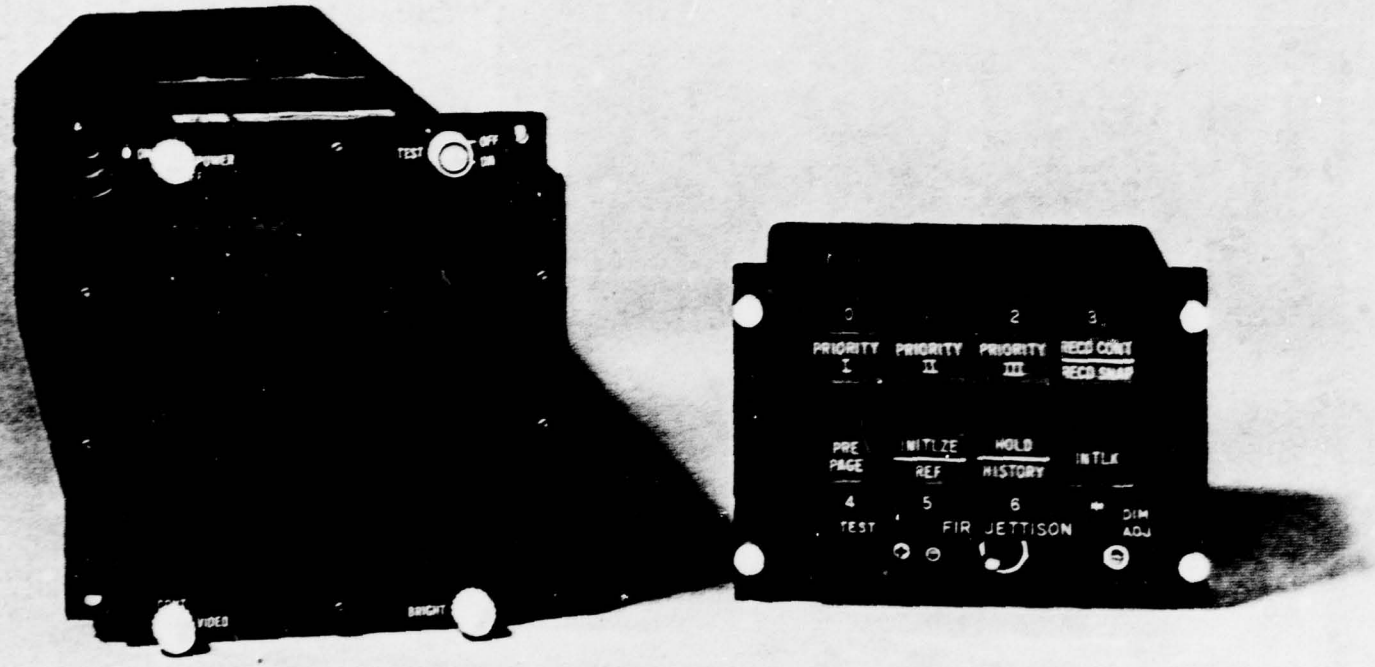
# ULAIDS SIGNAL ACQUISITION AND CONDITIONING TERMINAL (SACT)



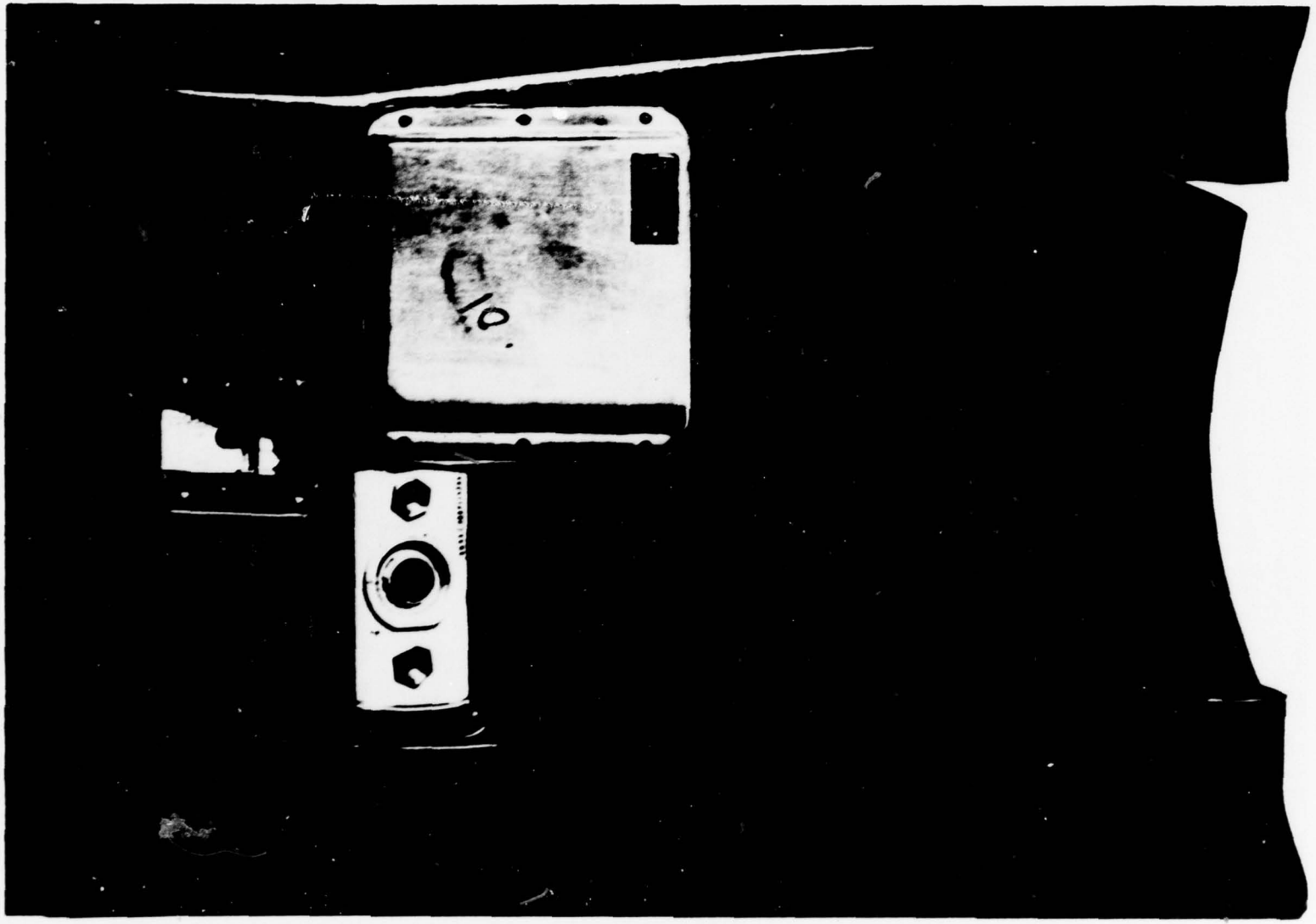
# ULAIDS AIRCRAFT HEALTH MONITOR RECORDER (AHMR)



# ULAIDS MASTER MONITOR DISPLAY/DATA ENTRY PANEL (MMD/DEP)



# ULAIDS FLIGHT INCIDENT RECORDER/UNIVERSAL LOCATOR (FIR/UL) DEPLOYABLE UNIT



# ULAIDS SUBSYSTEMS CHARACTERISTICS

## AN/AYQ-8 (V) (XN-1)

| COMPONENT | ATR SIZE | SIZE ($IN^3$) | WEIGHT (LBS) | | POWER (WATTS) | |
|---|---|---|---|---|---|---|
| | | | XN-1 | (PRODUCTION) | XN-1 | PRODUCTION |
| SACT (2) | 3/4 LONG | 1578 | 46 | (32) | 180 | (112) |
| MMD | 3/4 LONG | 544 | 16 | (16) | 68 | (68) |
| FIR/UL | — | 1090 | 10 | (8) | 20 | (10) |
| AHMR | 1 LONG | 833 | 23 | (17) | 82 | (54) |
| DEP | 1/4 SHORT | 75 | 3 | (2) | 3 | (2) |
| MT (2) | 3/4 SHORT | 1068 | 27 | (8) | 135 | (84) |
| CABLES/SENSORS | | — | 6 | (6) | — | — |
| TOTAL SYSTEM | | 5121 | 131 | (89) | 488 | (330) |

# ULAIDS SOFTWARE DESIGN

- MODULAR AND FLEXIBLE
- STANDARDIZED EXECUTIVE SOFTWARE THAT CAN BE INDEPENDENTLY CONFIGURED FOR EACH SUBSYSTEM DISTRIBUTED MICROCOMPUTER
- MODULAR EXECUTIVE SOFTWARE DESIGN THAT IS TABLE DRIVEN WHEREVER POSSIBLE ALLOWING MAXIMUM FLEXIBILITY TO INCLUDE OR EXCLUDE FEATURES AS REQUIRED
- APPLICATION SOFTWARE RUNS AS INDEPENDENT TASKS, SUBORDINATE TO AND SUPERVISED BY THE EXECUTIVE SOFTWARE. SUBSYSTEM TASK LISTS ARE GENERATED AT CONFIGURATION (ASSEMBLY TIME) AND ARE EASILY MODIFIED TO ADD OR DROP TASKS.

# ULAIDS DATA
# DISPLAY/RECORD PRIORITIES

| PRIORITY | DESTINATION | FLIGHT CONDITION |
|---|---|---|
| I | MMD<br>FIR<br>AHMR | CATASTROPHIC, SAFETY OF FLIGHT OR MISSION ABORT |
| II | MMD<br>FIR<br>AHMR | ADVISORY, SUBSYSTEM FAILURE OR DEGRADED PERFORMANCE |
| III | AHMR | AIRCRAFT/ENGINE SUBSYSTEM HEALTH |

# ULAIDS SENSOR/PARAMETER SIGNAL CAPACITY

| SIGNAL TYPE | TYPICAL MIN. CAPACITY | TYPICAL MAX. CAPACITY |
|---|---|---|
| DISCRETE (0 & 5 VDC LEVELS) | 4 | 12 |
| DISCRETE (0 & 28 VDC LEVELS) | 24 | 72 |
| DISCRETE (CLOSURE TO GND) | 8 | 24 |
| PULSE COUNTER/TORQUER (0 & 5 VDC LEVELS) | 1 | 2 |
| AC ANALOG (DIFFERENTIAL) | 10 | 10 |
| DC ANALOG (DIFFERENTIAL) | 14 | 14 |
| SYNCRO | 8 | 16 |
| SERIAL DIGITAL | 2 | 2 |
| MAGNESYN | 4 | 4 |
| TACHOMETER | 4 | 4 |
| FREQUENCY | 4 | 4 |
| THERMOCOUPLE | 3 | 3 |
| RESISTANCE THERMOMETER | 4 | 4 |
| RESOLVER | 8 | 8 |
| PIEZOELECTRIC ACCELEROMETER | 8 | 8 |
| TOTAL PER SACT | 106 | 187 |
| TOTAL (TWO SACTS) | 212 | 374 |

# ULAIDS MEMORY CAPACITY (WORDS)

| SUBSYSTEM | PROTOTYPE CONFIGURATION | | MAXIMUM CAPACITY | |
|---|---|---|---|---|
| | PROM | RAM | PROM | RAM |
| SACT (2) | 48K | 12K | 72K | 32K |
| AHMR | 6K | 6K | 24K | 16K |
| FIR/UL | 6K | 6K | 24K | 16K |
| MMD/DEP | 36K | 8K | 48K | 16K |
| TOTAL | 96K | 32K | 168K | 80K |




# CAPABILITIES



| | |
|---|---|
| AIRBORNE/REAL TIME: | MMD/DEP |
| POST FLIGHT ANALYSIS: | AHMR/GTU |
| POST INCIDENT ANALYSIS: | FIR/UL/GTU |

# GROUND TERMINAL UNIT (GTU)



# AIRBORNE/REAL TIME

- PARAMETER LIMIT EXCEEDANCE OR FAILURE - AUTOMATIC ALERT DISPLAY AND RECORD
- DEGRADING TRENDS - AUTOMATIC ADVISORY DISPLAY AND RECORD
- EMERGENCY PROCEDURES - AUTOMATIC ALERT DISPLAY
- SELECTED CURRENT PARAMETERS (GAUGE SUBSTITUTION) - AUTOMATIC OR CALLUP DISPLAY
- FLIGHT LINE MAINTENANCE - CALLUP DISPLAY (HISTORY OF PARAMETER LIMIT EXCEEDANCE WITH FAULT ISOLATION)

# SAFETY

- CRASH ANALYSIS - FLIGHT INCIDENT RECORDER
- CRASH POSITION LOCATION - UNIVERSAL LOCATOR
- SYSTEMS SAFETY - HEALTH MONITORING

# POST FLIGHT ANALYSIS

- IN DEPTH MAINTENANCE DIAGNOSTICS
- PERFORMANCE TRENDING
  - TOTAL SYSTEM EFFICIENCY OPTIMIZATION THROUGH SUB-SYSTEM TRIMMING/ADJUSTMENT — ENGINE FUEL CONTROL, DRAG, LANDING PERFORMANCE
  - SINGLE POINT DEGRADATION THROUGH SUB-SYSTEM PERFORMANCE MONITORING AND EVALUATION - ACCURACY, CONSISTANCY, TIMELINESS
  - PILOT PERFORMANCE/TRAINING
- FORECASTING
  - FAILURE PROJECTION & REPLACEMENT
  - ELIMINATE PHASE AND FORCED REMOVAL MAINTENANCE CONCEPTS

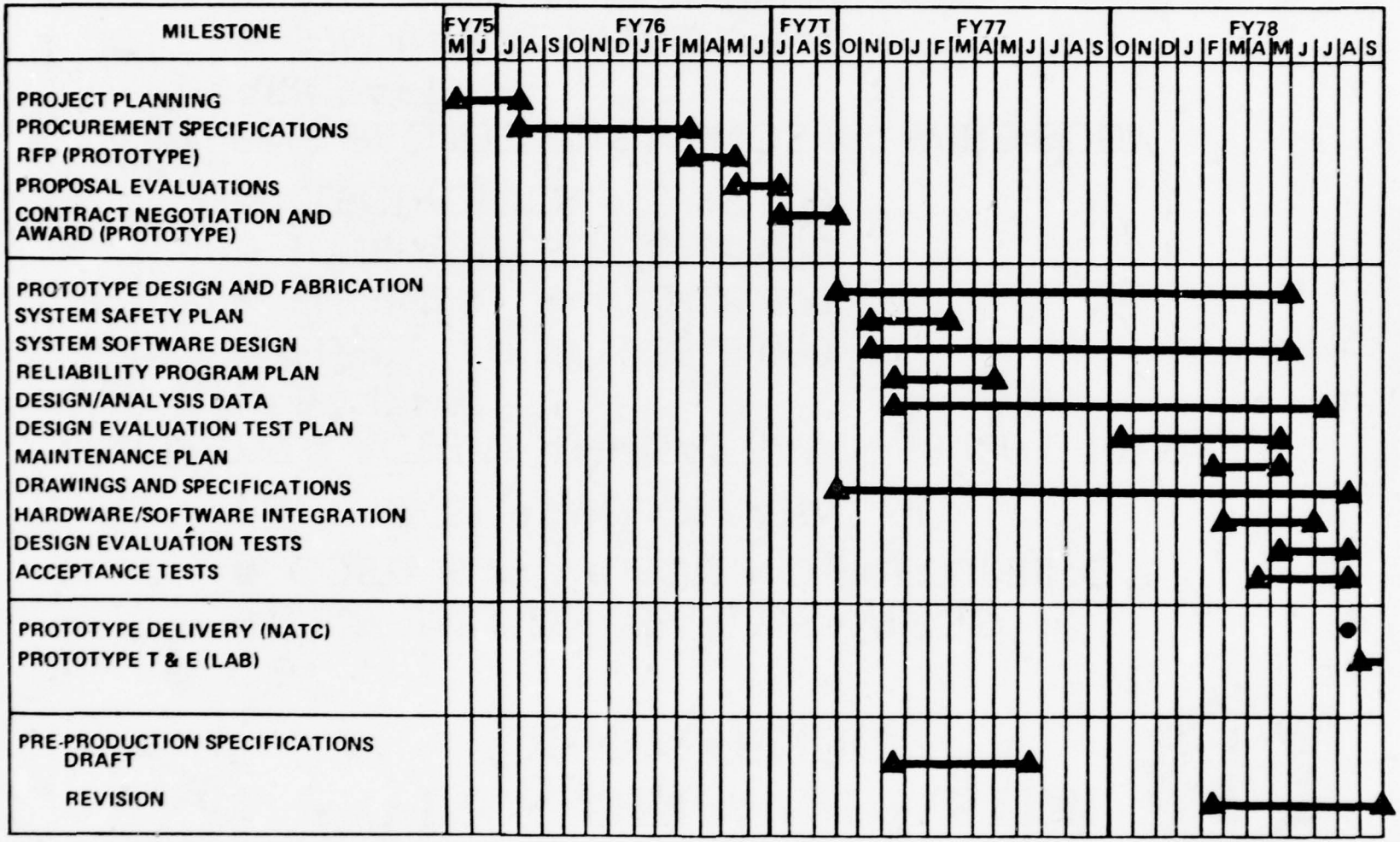
# RELATED PROGRAMS

- ADVANCED INTEGRATED DISPLAY SYSTEMS (AIDS) — NAVY
- DIGITAL AVIONICS INFORMATION SYSTEMS (DAIS) — AIR FORCE
- INTEGRATED ENGINE INSTRUMENT SYSTEM (IEIS) — NAVY
- GENERAL PURPOSE MULTIPLEXING SYSTEM (GPMS) — NAVY
- INFLIGHT ENGINE CONDITION MONITORING SYSTEM (IECMS) — NAVY
- AUTOMATIC INSPECTION DIAGNOSTIC AND PROGNOSTIC SYSTEM (AIDAPS) — ARMY
- BASIC AVIONIC SUBSYSTEM INTEGRATION CONCEPT (BASIC LAB) — NAVY

# ADVANTAGES

- QUANTUM IMPROVEMENTS IN OVERALL AIRCRAFT
  - MISSION EFFECTIVENESS
    - STANDARDIZED REAL-TIME SYSTEM MANAGEMENT AND BACKUP LOGIC
    - INCREASED SUCCESS CONFIDENCE (FROM PROJECTION AND TRENDING CAPABILITIES)
  - SAFETY
    - OVERALL SYSTEM SAFETY
    - UNIVERSAL LOCATOR/INCIDENT ANALYSIS
  - RELIABILITY
    - CONTINUOUS SYSTEM HEALTH STATUS
    - SYSTEM CONFIDENCE IN ABSENCE OF PROJECTED FAILURES OR ADVERSE TRENDS
  - ILS & MAINTAINABILITY
    - ACCURATE AND TIMELY MAINTENANCE ACTION
    - FAILURE PROJECTION/SYSTEM TRENDING
      - LOWER MAINTENANCE HOURS
      - LOWER MATERIAL COSTS

# ULAIDS MILESTONE CHART



# SAVINGS THROUGH SYSTEM MONITORING

- C-5
  - $5.65 MILLION YEARLY SAVINGS
  - COST: $.79 MILLION NON-RECURRING
    $1.9 MILLION ANNUAL
- TWA ACTUAL ANNUAL SAVINGS (30 L-1011 & 10 B-747)
  - FUEL $343,000
  - ENGINES $428,000
  - HYDRAULICS $130,000
  - ELECTRONICS $ 9,000
- TWA PROJECTED SAVINGS (30 WIDE-BODY AIRCRAFT)
  - $9M INITIAL INVESTMENT
  - $1M ANNUAL RECURRING COST
  - $6.5M ANNUAL SAVINGS BEYOND 1ST YEAR
  - PAYBACK IN 2.25 YEARS (INCLUDING RECURRING COSTS)

# POTENTIAL
# NAVY SAVING AREAS

| | |
|---|---|
| ● AIRCRAFT INCIDENTS (MATERIAL FAILURE PRECIPITATED) | $100 MILLION YEARLY |
| ● MAINTENANCE: | |
| ● NON-DEFECTIVE REPAIR (A-799 RATE) | $10 MILLION YEARLY |
| ● SCHEDULED MAINTENANCE (PHASE) | |
| ● SPARE PARTS LOGISTICS | |
| ● PROCUREMENT: | |
| ● ULAIDS VS PECULIAR DESIGN FOR EACH NEW AIRCRAFT | SAVINGS TO BE DETERMINED |

**ELECTRONIQUE MARCEL DASSAULT**
**LABORATOIRES ET USINE**
55, Quai Carnot - 92214 SAINT-CLOUD 602.50.00

SECOND INTERNATIONAL MULTIPLEX DATA
BUS CONFERENCE

10 , 11 , 12 , OCTOBER 1978
DAYTON (OHIO)

# GINA AIRBONE MULTIPLEX DIGITAL DATA BUS

by M. Alain QUIDET

ELECTRONIQUE MARCEL DASSAULT - FRANCE -

PRECEDING PAGE BLANK

E.M.D. 40.657-M ST-CLOUD, le 20 Septembre 1978.-



Ce document est la propriété de l'Electronique Marcel Dassault et ne peut être reproduit ou communiqué sans autorisation écrite.

SUMMARY

CHAPTER I : INTRODUCTION

CHAPTER II : GENERAL

2.1. DEFINITIONS

2.1.1. Bus line or "bus"
2.1.2. Bus controller (UG)
2.1.3. Remote terminal (P)
2.1.4. Couplers
2.1.5. Digital transmission system

2.2. OPERATIONAL SAFETY

CHAPTER III : TRANSMISSION SUPPORT

3.1. TRANSMISSION SUPPORT STRUCTURE

3.1.1. Type of structure
3.1.2. Connecting mode
3.1.3. Liaison type

3.2. INTER-EQUIPMENT LIAISON CONCEPT

3.2.1. Transmission support construction
3.2.2. Electrical characteristics of line
3.2.3. Adaptation
3.2.4. Shielding
3.2.5. Bus topology

3.3. CHARACTERISTICS OF TRANSMISSION SIGNALS

3.3.1. Bit rate
3.3.2. Modulation and code type
3.3.3. Electrical characteristics



Ce document est la propriété de l'Electronique Marcel Dassault et ne peut être reproduit ou communiqué sans autorisation écrite.

CHAPTER IV : SYSTEM OF EXCHANGES

4.1. INFORMATION ELEMENTS TRANSMITTED

4.1.1. Character
4.1.2. Chain of characters
4.1.3. Word
4.1.4. Message
4.1.5. Informational content

4.2. EXCHANGE EXECUTION

4.2.1. Exchanges
4.2.2. Peripheral synchronisation
4.2.3. Exchange procedure

4.2.3.1. Exchange structure
4.2.3.2. Bus configuration and equipment preselection

4.2.3.2.1. Command word chain structure
4.2.3.2.2. Sub-buses connection and peripheral addressing

4.2.3.3. Exchange execution

4.2.3.3.1. Succession of phases
4.2.3.3.2. Command execution
4.2.3.3.3. Data emission

4.3. ADDRESSES OF THE EQUIPMENT

CHAPTER V : EXCHANGE SYSTEM MANAGEMENT

5.1. EXCHANGE MODES
5.2. DATA EXCHANGE

CHAPTER VI : EXCHANGE SYSTEM CHECK

6.1. TRANSMISSION CHECK
6.2. INTEGRITY CHECKS OF EXCHANGE SYSTEM

6.2.1. Checks carried out by the UG'S
6.2.2. Specific checking devices



Ce document est la propriété de l'Electronique Marcel Dassault et ne peut être reproduit ou communiqué sans autorisation écrite

# ILLUSTRATIONS LIST

ILLUSTRATION 1

DIGITAL TRANSMISSION SYSTEMS

ILLUSTRATION 2

REDUNDANCY EXAMPLE

ILLUSTRATION 3

STRUCTURE OF AN EXCHANGE

ILLUSTRATION 4

COMMAND WORDS F FIELD CODING

ILLUSTRATION 5

GINA Bus Hardware

ILLUSTRATION 6

Bus Interface Card

ILLUSTRATION 7

Bus Tester

ILLUSTRATION 8

GINA Bus Integration Test Station



Ce document est la propriété de l'Electronique Marcel Dassault et ne peut être reproduit ou communiqué sans autorisation écrite.

## CHAPTER I — INTRODUCTION

ELECTRONIQUE MARCEL DASSAULT are working since 1970 in the field of multiplex data buses at 1 MHz.

Thus ELECTRONIQUE MARCEL DASSAULT have successively studied the following multiplex information transfer systems through contracts under French Authorities :

- 1970 - 1973 : bus for Missiles,
- 1973 - 1976 : GINA bus for Aircraft,
- 1975 - 1977 : directly derived from GINA bus , a bus for NAVY applications.

Such studies now concretised by several operational applications, as the recent French military programs, issued by D.T.C.A., D.T.E.N. and D.C.A.N., are all using the GINA bus for transfer of information between equipment.

The purpose of this statement is to explain some essential characteristics of the GINA bus.



Ce document est la propriété de l'Electronique Marcel Dassault et ne peut être reproduit ou communiqué sans autorisation écrite

# CHAPTER II

# GENERAL

## 2.1. DEFINITIONS

### 2.1.1. Bus line or "bus"

This is the transmission support for digital information from a multiplex exchange system.

### 2.1.2. Bus controller (UG)

This unit insures organization of exchanges on the bus in autonomous (local mode) or in liaison with one of the system computer (channel mode). It must be connected to the main bus.

### 2.1.3. Remote terminal (P)

Each remote unit connected with the bus line comprises :

- the equipment itself,
- the couplers as necessary for connection of the equipment to the bus line.

### 2.1.4. Couplers

Is the name given to a functional assembly being a part of each remote terminal and is generally integrated in each unit ; it realises the connection of the equipment itself with the bus line.

Such assembly is composed of :

- one or several standard bus coupler (s) (C.O.S.), according with the redundancy

  The bus coupler is the element insuring connection with the bus line ; its definition (except for redundancy) and the functions it performs are identical for all equipment. It is only a function of the definition of the transmission system and exchanges organization on the bus. It carries out all logic operations that must perform all units connected onto a same bus.

- a sub-system coupler (CSS)

  The sub-system coupler perform the functions bound to the transmission system which are not included in the bus coupler (s) and thus, specific to the requirements of the equipment into which it is integrated.

  Physically, the sub-system coupler insures the coupling of the equipment itself with the bus coupler (s).



Ce document est la propriété de l'Electronique Marcel Dassault et ne peut être reproduit ou communiqué sans autorisation écrite

2.1.5. Digital transmission system

The basic transmission system comprises (Illustration 1 - Figure 1) :

- the bus-line (redundant if necessary),
- the bus-controller (redundant if necessary),
- the standard bus couplers for each equipment.

From the basic transmission system, some sub-buses may be connected to the main bus (see Figure 2 - Illustration 1).

Other combinations may be also used.

## 2.2. OPERATIONAL SAFETY

Redundancy allows for a good safety of information exchanges and for failure protection.

The redundancy degree is determined at each level as a function of specific requirements for the general system. For example (Ill. 2), the redundancy could bare upon the controller and the main bus, all sub-buses being simple.

In all cases, the couplers COS and CSS are considered as being integral part of the equipment and have not to be redundant if the equipment itself is not redundant.



Ce document est la propriété de l'Electronique Marcel Dassault et ne peut être reproduit ou communiqué sans autorisation écrite.

CHAPTER III

# TRANSMISSION SUPPORT

## 3.1. TRANSMISSION SUPPORT STRUCTURE

### 3.1.1. Type of structure

Equipment connection on the bus line (peripheral P) is made by stubbing , the main line being not interrupted by the connection of a peripheral.

### 3.1.2. Connecting mode

Liaison between equipment and the bus is made through transformers with a stubbing.

### 3.1.3. Liaison type

The digital information exchange system is organized around two lines :

- one procedure line (LP) is unidirectional , on which the controller emits the procedure words destined to equipment,
- one data line (LD) bidirectional on which date (emission and reception) are transmitted.



Ce document est la propriété de l'Electronique Marcel Dassault et ne peut être reproduit ou communiqué sans autorisation écrite.

## 3.2. INTER-EQUIPMENT LIAISON CONCEPT

### 3.2.1. Transmission support construction

The cable used is a twisted pair with double shielding, closed at both ends on its characteristic impedance.

### 3.2.2. Electrical characteristics of line

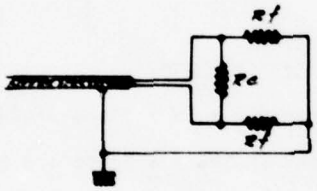
The line used must have a characteristic impedance :

Zc of 75Ω ± 5%

Attenuation caused by a 100 m length of cable on a sinusoïdal signal at 1 MHz must be less than 3 dB.

### 3.2.3. Adaptation

At both ends, the line is fitted with the following network:



$R_c$ = 75Ω ± 1% 0,5 W

$R_f$ = 10KΩ ± 5% 0,25W.

### 3.2.4. Shielding

At each point where the shielding is interrupted (equipment connection, bus termination, bulkhead crossing, etc...), it must be connected to framework by means of as short as possible a liaison. Only the two wires of the line should pass through the wall, the ground liaison being made from outside.



### 3.2.5. Bus topology

For definition of the bus, the following requirements must be strictly observed :

- The line portion between two consecutive stubbs must be longer than the sum of the stubbs length.
- The length of a stubb, measured from the main bus to the transformer of the equipment must not exceed 1,5 m.
- On the bus, there must not exist any way longer than that necessary to go from one adaptation network to the other.



Ce document est la propriété de l'Electronique Marce[illegible] lit et ne peut être reproduit ou communiqué sans autorisation écrite.

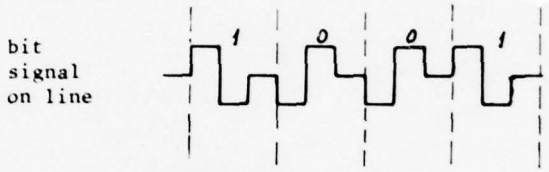
## 3.3. CHARACTERISTICS OF TRANSMISSION SIGNALS

### 3.3.1. Bit rate

Informations are transmitted under differential mode in series on the lines at the rate of 1 megabit per second.

### 3.3.2. Modulation and code type

The code used is a biphase code with return to zero and thus, modulation is at three levels. The continuous component, null at the bit level, allows the use of transformers in good conditions.




### 3.3.3. Electrical characteristics

- Emitting levels : ± 6V ± 1V on 37.5Ω
- Reception thresholds : ± 2.25V ± 0.25V
- Common mode rejection :

In the frequency band 0 to 2 MHz, any signal of a peak amplitude inferior or equal to ± 25V between line and ground does not impair the liaison performances. In addition, any signal of peak amplitude inferior or equal to ± 50V is not liable to dammage definitively the associated equipment.



Ce document est la propriété de l'Electronique Marcel Dassault et ne peut être reproduit ou communiqué sans autorisation écrite.

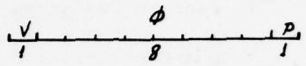
# CHAPTER IV SYSTEM OF EXCHANGES

## 4.1. INFORMATION ELEMENTS TRANSMITTED

### 4.1.1. Character

A character is an information element composed of 10 inseparable bits at transmission.

These 10 bits are grouped according to the following format:




The V bit signification depends upon the type of the character (procedure, echo, data) in which it is included.

The field Ø contains the useful byte.
The P bit is an imparity bit bearing on the whole character (odd number of "1" or "0").

### 4.1.2. Chain of characters

The characters may be grouped in sequence to form chains of 1 to 256 characters.

### 4.1.3. Word

The characters may be grouped to form words. The data word format transmitted is free.

### 4.1.4. Message

A message is an assembly of characters carried on a line as a unique block (without "gap").

### 4.1.5. Informational content

When an information is coded in several bit contained in one or more fields of the useful byte these will be transmitted most significant bit ahead.

When an information is coded on several bytes the characters containing them are transmitted most significant bit ahead.



Ce document est la propriété de l'Electronique Marcel Dassault et ne peut être reproduit ou communiqué sans autorisation écrite.

## 4.2. EXCHANGE EXECUTION

### 4.2.1. Exchanges

An exchange corresponds to the ensemble of messages carried on the bus from one single initiation of the bus controller (U.G.).

It runs in two steps :

- Bus configuration (sub-buses connection) and pre-selection of those peripherals concerned by this exchange. This function is performed by the U.G. by means of the procedure line.

- Data transfer (if any) between preselected units (among which occasionally the U.G.) through the data line.

The exchange is monitored by the U.C. by means of the message it emit through the procedure line.

### 4.2.2. Peripheral synchronisation

The synchronisation of the peripherals is carried out :

- at the bit level : by the auto-sync modulation,

- at the character level : by counting the bits,

- at the message level : by envelope detection,

- at exchange level : by the messages included.

Any receiver utilizing a message controls :

- the imparity bits of the characters herein,

- its format (concrete number of 10 bits characters received).

### 4.2.3. Exchange procedure

#### 4.2.3.1. Exchange structure

Illustration 3 represents the messages circulating on L.P. and L.D. lines in execution of an exchange.

The procedure message, issued on LP by the U.G. is including :

- a chain of command words of two characters each (command phase),

- occasionnally , a chain of delaying characters (delay phase, if any),



Ce document est la propriété de l'Electronique Marcel Dassault et ne peut être reproduit ou communiqué sans autorisation écrite.

- occasionnally, a chain of service characters (data phase, if any),
- a service character for end of exchange *.

The successive data messages, emitted by equipment on L.D. are of two types :

- acknowledgement of pre-selection and status or echo from addressed elements (command phase), emitted on L.D. by equipment as designated by every command word,
- data message issued by the peripheral emitter, composed of :

  . one chain of data characters,

  . one data character for end of exchange*.

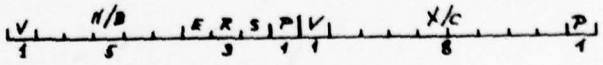
*

Remark : Eventually, when security of transmission would be mandatory, such service or data characters of end of exchange could contain a longitudinal imparity bearing on the whole procedure or data message.

4.2.3.2. Bus configuration and equipment preselection

Such operations are carried out during the command phase by the command words issued on L.P. by the controller U.G.

The command words have the following format :




- When F is different from "0" :

  The first character (most significant bits) selects one peripheral, explicitely designated by the five bits in the N field and it defines the function to be executed for the bus in the three bits of F field (operative code).
  The second character contains an X byte which is interpreted by the equipment as a function of code F.

- When F = "0" :

  The first character (most significant bits) selects a sub-bus coupler, explicitely designated by the five bits of B field.

  The second character contains an byte C which is interpreted by equipment formely preselected.

  Table on illustration 4 indicates the coding of F field and the different corresponding command words.



Ce document est la propriété de l'Electronique Marc ilt et ne peut être reproduit ou communiqué sans autorisation écrite

4.2.3.2.1. Command word chain structure

The command phase could be separated in successive sub-phases, corresponding to the various sub-buses involved in the exchange. During each sub-phase , the controller U.G. emits on L.P. a sub-chain of command words destined to equipment connected to the corresponding sub-bus.

Each sub-chain comprises :

- the command words destined to the peripherals of the sub-bus (if any),

- one additionnal command and connection word.

4.2.3.2.2. Sub-buses connection and peripheral addressing

Connection of a sub-bus is carried out by a sub-bus coupler CSB addressed by a CC .

On a sub-bus, no information is circulating prior to the start of the corresponding command sub-phase. For an equipment connected to this sub-bus, the command sub-phase begins only when the message appears on L.P. and it terminates at the end of receipt of the first CC .

Remark : An equipment ignores it is connected on a sub-bus; for it, the command phase, limited to the sub-phase on its sub-bus, is exactly the same as for a single bus, without any sub-bus.

4.2.3.3. Exchange execution

4.2.3.3.1. Succession of phases

An exchange is displayed into three main phases :

- Command phase.

- Occasionnally , delay phase.

- Occasionnally , data phase.

An end of exchange phase, not seen by equipment , is always associated to these three phases at the transmission level (see parag. 4.2.3.1.).

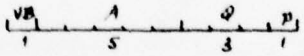


Ce document est la propriété de l'Electronique Marcel Dassault et ne peut être reproduit ou communiqué sans autorisation écrite.

4.2.3.3.2. Command execution

When an equipment (CSB or peripheral) is explicitely addressed by a command (,ield N/B), it emits on L.D. towards U.G. a message of one preselection acknowledgement character or "echo" (ill. 3).

This echo has the following format :




It contains :

- Address A of the equipment or CSB,
- a status Q of the peripheral or CSB, transmitted to the controller U.G.

The VB bit indicates the validity of the bus seen by the equipment (result of checks made by the COS or the CSB on the transmission since the beginning of the preceding exchange it has seen, even when not explicitely addressed itself).

To issue this echo, the equipment utilises the clock extracted from the message received on L.P.

4.2.3.3.3. Data emission (ill. 3)

When a peripheral has been preselected as emitter during the command phase, it emits on L.D. (during the data phase) a chain of as many characters as in the message emitted on LP by the U.G. there are bits V = 1,.At the end of this character chain, the COS introduces a data character of end of exchange (end of exchange phase).

In order to issue this chain of characters, the peripheral utilises the clock extracted from the message received on L.P.



Ce document est la propriété de l'Electronique Marcel Dassault et ne peut être reproduit ou communiqué sans autorisation écrite

## 4.3. ADDRESSES OF THE EQUIPMENT

Attribution of address to equipment (peripheral or CSB) must be done as a function of the exchange system ensemble and of its management. The following rules must be followed due to the procedure :

- on a given sub-bus , all peripherals must have different addresses,
- address N = 0 is prohibited for CSB and peripherals,
- during an exchange , the bus configuration must never be such that several CSB of same address (if any) could be simultaneously addressed.



Ce document est la propriété de l'Electronique Marcel Dassault et ne peut être reproduit ou communiqué sans autorisation écrite.

# CHAPTER V EXCHANGE SYSTEM MANAGEMENT

## 5.1. EXCHANGE MODES

Execution of an exchange may imply two types of action :

- Immediate action of U.G. on peripherals by means of command words.
- Data transfer between peripherals (UG included) subordinated to existence of a data phase.

  Both types of action are independently controlled by the UG and may be simultaneous (content of the command phase and issue of bits V = 1 generating the data phase).

The following two concepts allow, for each type of action, to classify the exchanges :

- The addressing mode of equipment can be :
  - explicit : N field of command words,
  - implicit :
    - . Label (broadcast mode) decoding for peripherals.
    - . Internal operation for U.G. controller.
- Exchange management mode can be :
  - centralized : UG is implicitely addressed and exchanges take place between UG and peripherals. UG does not emit echo on the bus.
  - decentralized : Exchanges take place between peripherals and when UG is concerned it is explicitely addressed as a normal peripheral . It has then on address A and is issuing an echo on the bus.



Ce document est la propriété de l'Electronique Marcel Dassault et ne peut être reproduit ou communiqué sans autorisation écrite

## 5.2. DATA EXCHANGE

The data exchange is conditionned by :

- the command phase which defines the type of exchange ,
- the existence of the data phase (bits V = 1 on L.P.) which executes the transfer.

The UG controller can then decide to carry out or not the data transfer , independently from the command phase contents.

At this level , it appears necessary to define the following concepts relative to data exchange :

- the data addressing mode, which can be :
  - explicit (local) : X field of command words,
  - implicit (general) : Label (broadcast mode).
- The data exchanged type, which can be :
  - Informations, proper meaning.
  - Status data.



Ce document est la propriété de l'Electronique Marcel Dassault et ne peut être reproduit ou communiqué sans autorisation écrite.

# CHAPTER VI — EXCHANGE SYSTEM CHECK

At the level of the system of exchanges, two types of checks are made :

- transmission checks,
- exchange system integrity checks.

## 6.1. TRANSMISSION CHECK

As a general rule any equipment utilizing a message checks its integrity at the transmission level.

Transmission checks are as follows :

- Imparity checks of all character composing the message.
- Format check of message at the end of transmission . The total number of n bits received must correspond to a concrete number K of characters of 10 bits : n = 10K.
- Longitudinal imparity check if the message has it (see parag. 4.2.3.1.).

When an equipment detects a fault in a message transmission.

. It stops immediately all orders execution (including these already received without error) contained in that message (command execution , data transfer, etc...) and locks on stand by until the end of the running message.

. It memorises the error, until the end of the command phase which will be seen in the next exchange.

. When explicitely addressed in the next exchange it points out the error by placing a VB bit = 0 in its echo.



Ce document est la propriété de l'Electronique Marcel Dassault et ne peut être reproduit ou communiqué sans autorisation écrite

## 6.2. INTEGRITY CHECKS OF EXCHANGE SYSTEM

Such checks are performed by :

- the monitoring elements of the exchange system (UG),
- specific elements when security of exchanges is mandatory (check of bus configuration, etc...).

### 6.2.1. Checks carried out by the UG'S

Such checks depends upon the definition of the exchange system management. In order to realize these checks, the UG'S can utilize the following informations :

- Equipment echoes on which they are checking :
  - the VB bit which provides the results of transmission checks made by the corresponding equipment,
  - concordance of address A with that of the addressed equipment,
  - its integrity towards transmission (parity and format).
- Other exchange messages on which they can-independently from the transmission checks-perform complementary checks to be determined as a function of management and organisation (redundancy at bus level, management,etc..) of each system.

### 6.2.2. Specific checking devices

Such devices may be placed anywhere along the bus and only exist when the security of exchanges is mandatory.

The checks carried out are determined as a function of the security requirements for each system.

Some possible local checks are quoted below :

- transmission checks,
- true addressing check of peripherals (concordance of echo with procedure message),
- bus configuration check.



Ce document est la propriété de l'Electronique Marcel Dassault et ne peut être reproduit ou communiqué sans autorisation écrite.

The two first types of checks are similar to those performed by the UG'S and may justify direct actions onto the managing units.

The last type of check allows, at sub-bus level, to insure of its good connection.

For this, the checking device is connected to the sub-bus : it verifies whether the A address contained in the echo issued by the CSB after execution of a connection corresponds really to the sub-bus on which it is connected. In the event of a detected error, this device locks out immediately the corresponding CSB.



Ce document est la propriété de l'Electronique Marcel Dassault et ne peut être reproduit ou communiqué sans autorisation écrite.

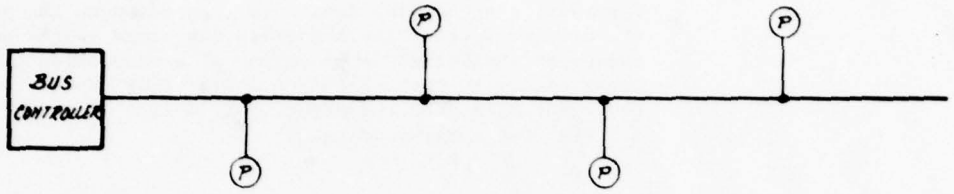



Fig. 1 . SIMPLE SYSTEM

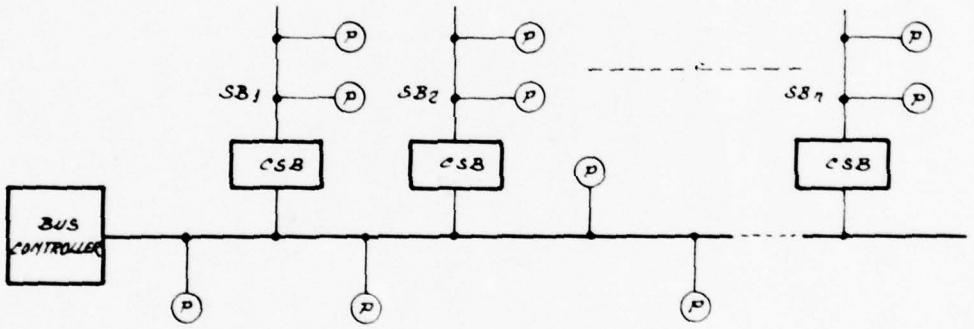



Fig. 2 . PARALLEL SYSTEM

ILLUSTRATION - 1

# DIGITAL TRANSMISSION SYSTEMS



44 3250

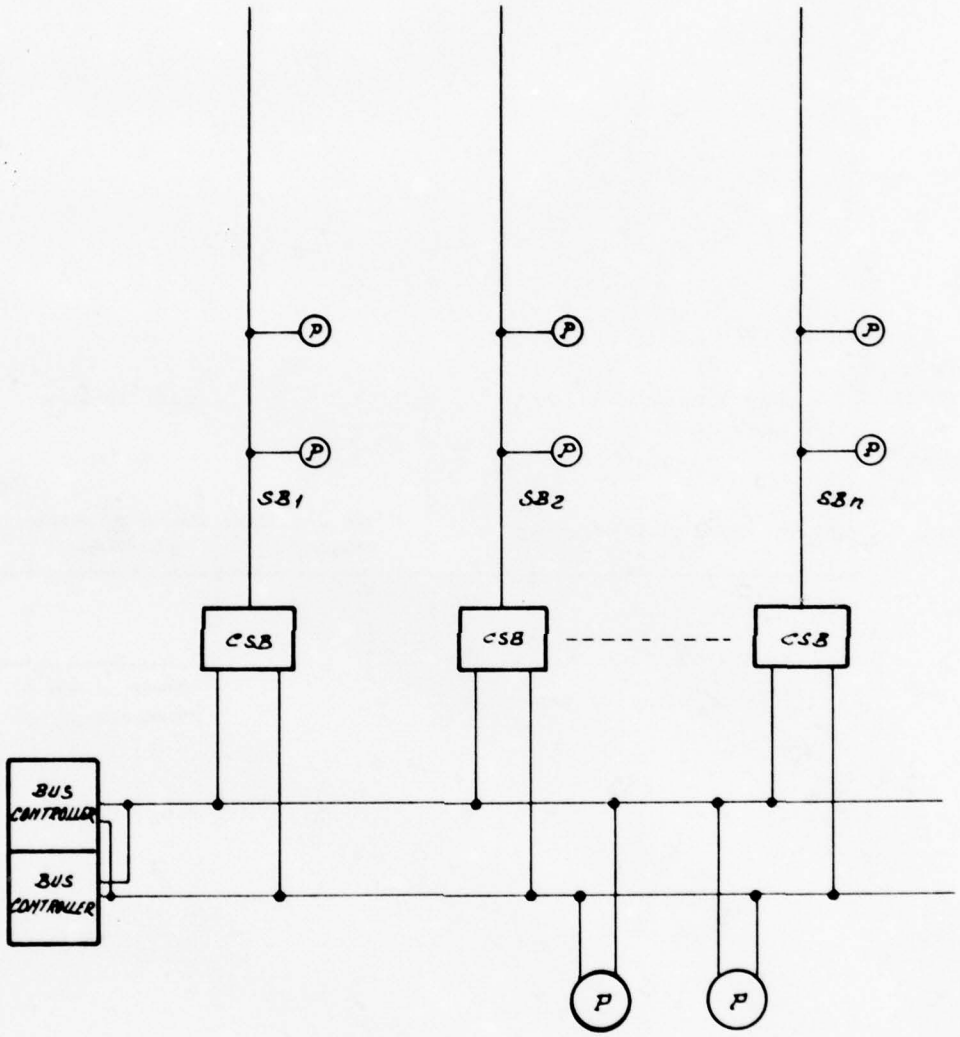



ILLUSTRATION - 2.

# REDUNDANCY EXAMPLE

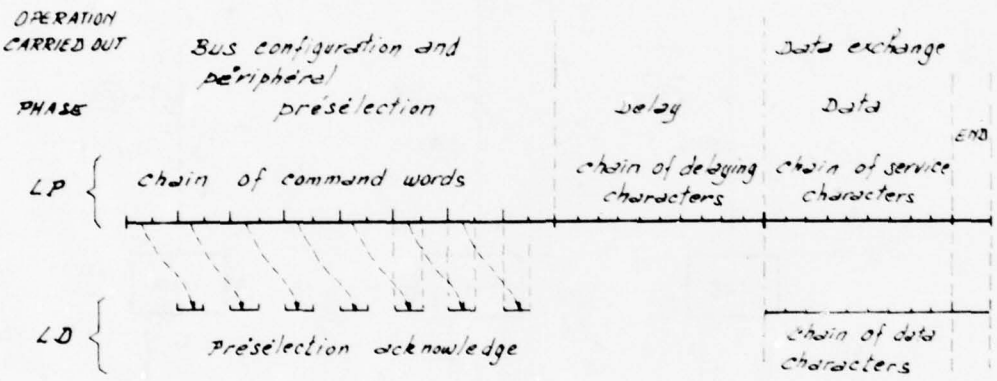
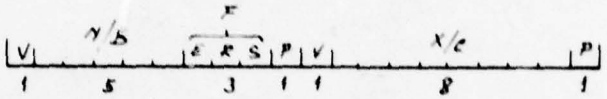



ILLUSTRATION. 3.

STRUCTURE OF AN EXCHANGE

## ILLUSTRATION. 4

### COMMAND WORDS F FIELD CODING




| F E.R.S. | Mnemonic Code | Command | Execution |
|---|---|---|---|
| 000 | CC | Connection and additional command | -CSBβ Connection<br>-Command complement (CC) for formerly preselected peripherals |
| 001 | CF | Function Command | -Immediate trigger of function X for peripheral N. |
| 010 | CR | Reception command | -Reception on I.D of data of local address X by peripheral N. |
| 011 | CS | Service command | -Reception on I.P of data of local address X by peripheral N. |
| 100 | CE | Emission command | -Emission on I.D of data of local address X by peripheral N. |
| 101 | CV | Channel command | -Emission on I.D of status of local address X by peripheral N. |
| 110 | CI | Label command | -Interpretation by peripheral N of this CI instead of CE or CV : emission command of general address X information.<br>-Interpretation by all other peripherals as a function of label X from this command CI as CR (according to labels |
| 111 | CT | Transfer command | Reception on IP and emission on ID of address X informations by peripheral N (simultaneous execution of CS and CE). |

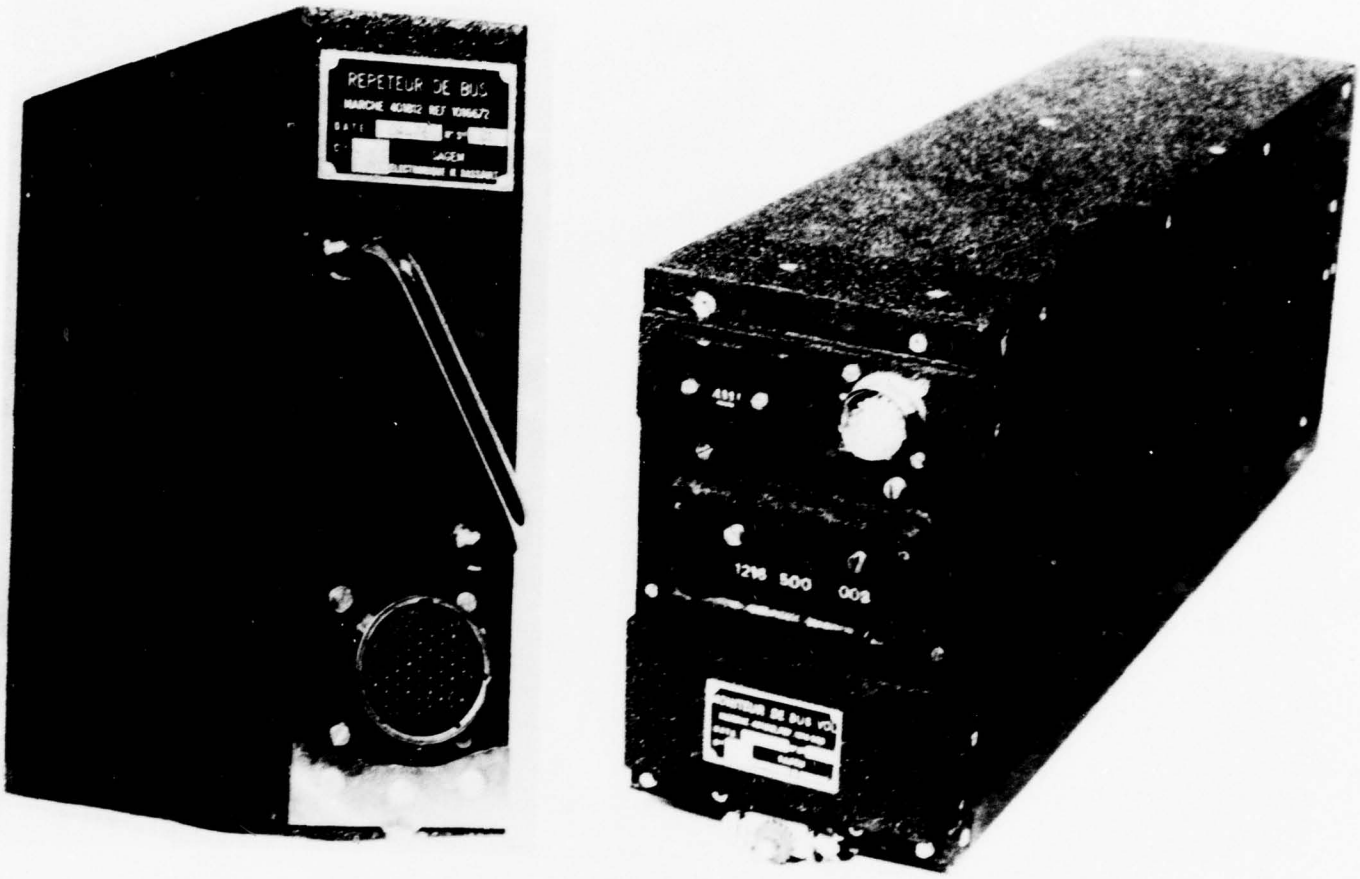



ILLUSTRATION 5. GINA Bus Hardware

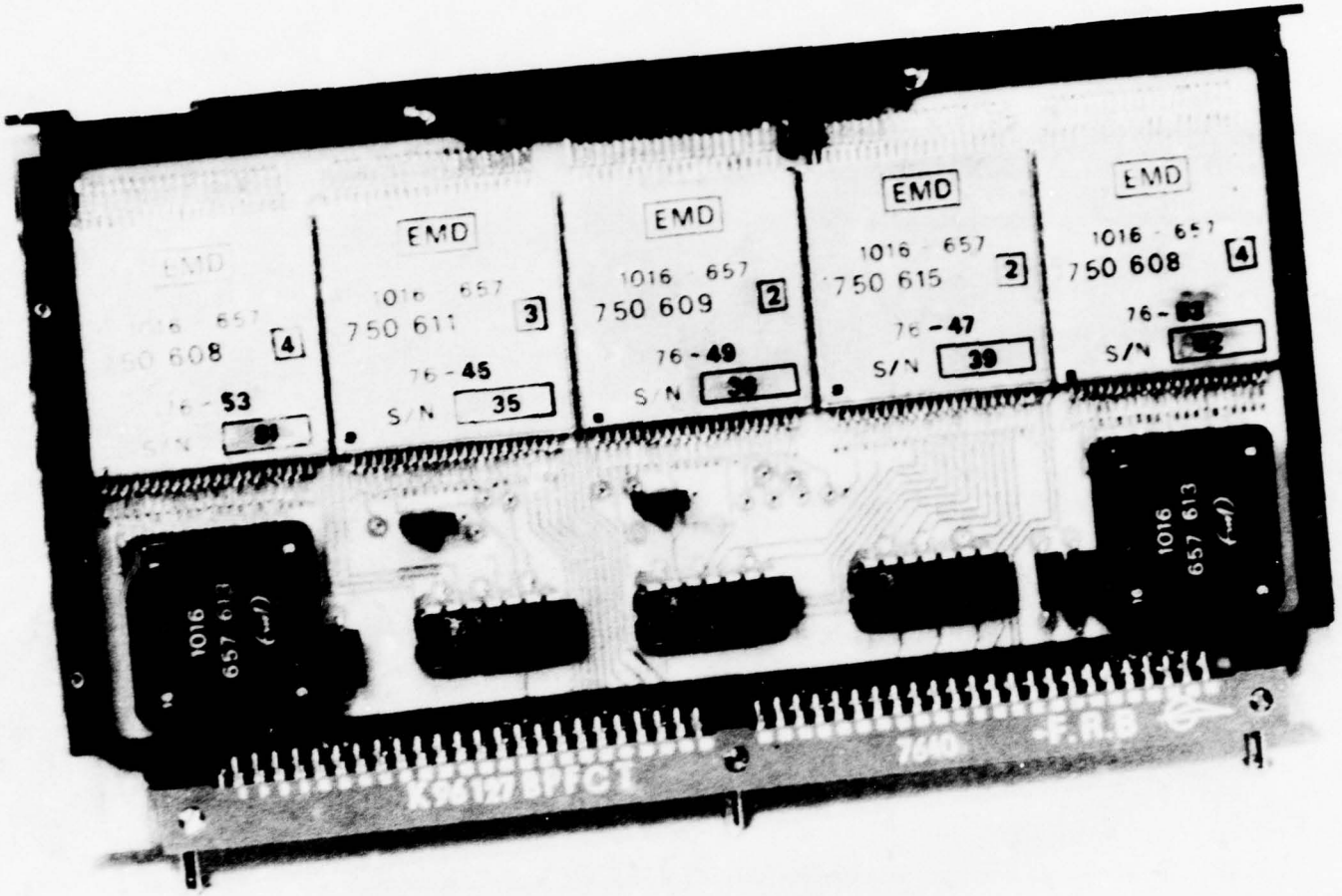



ILLUSTRATION 6. Bus Interface Card

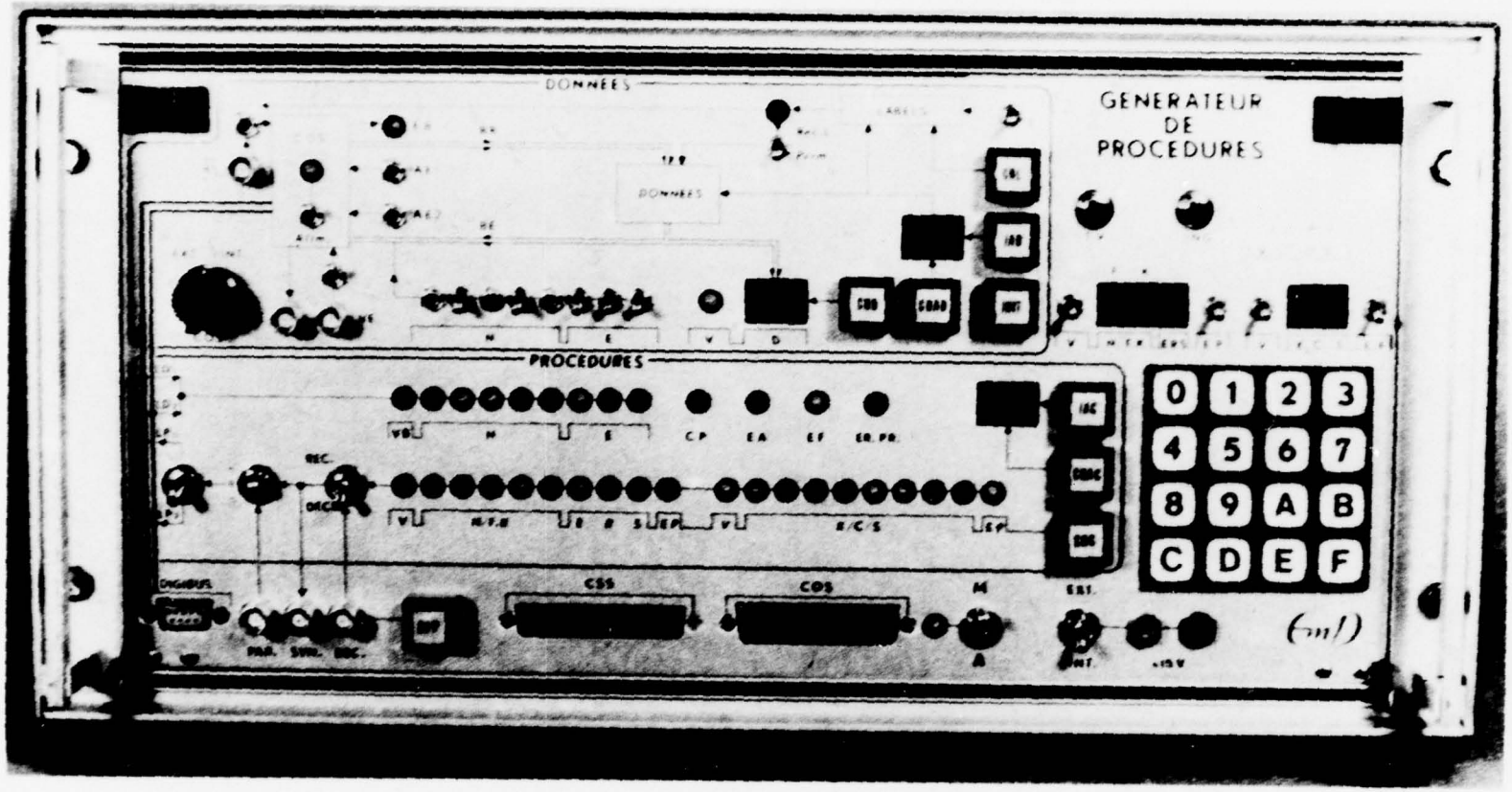



ILLUSTRATION 7. Bus Tester

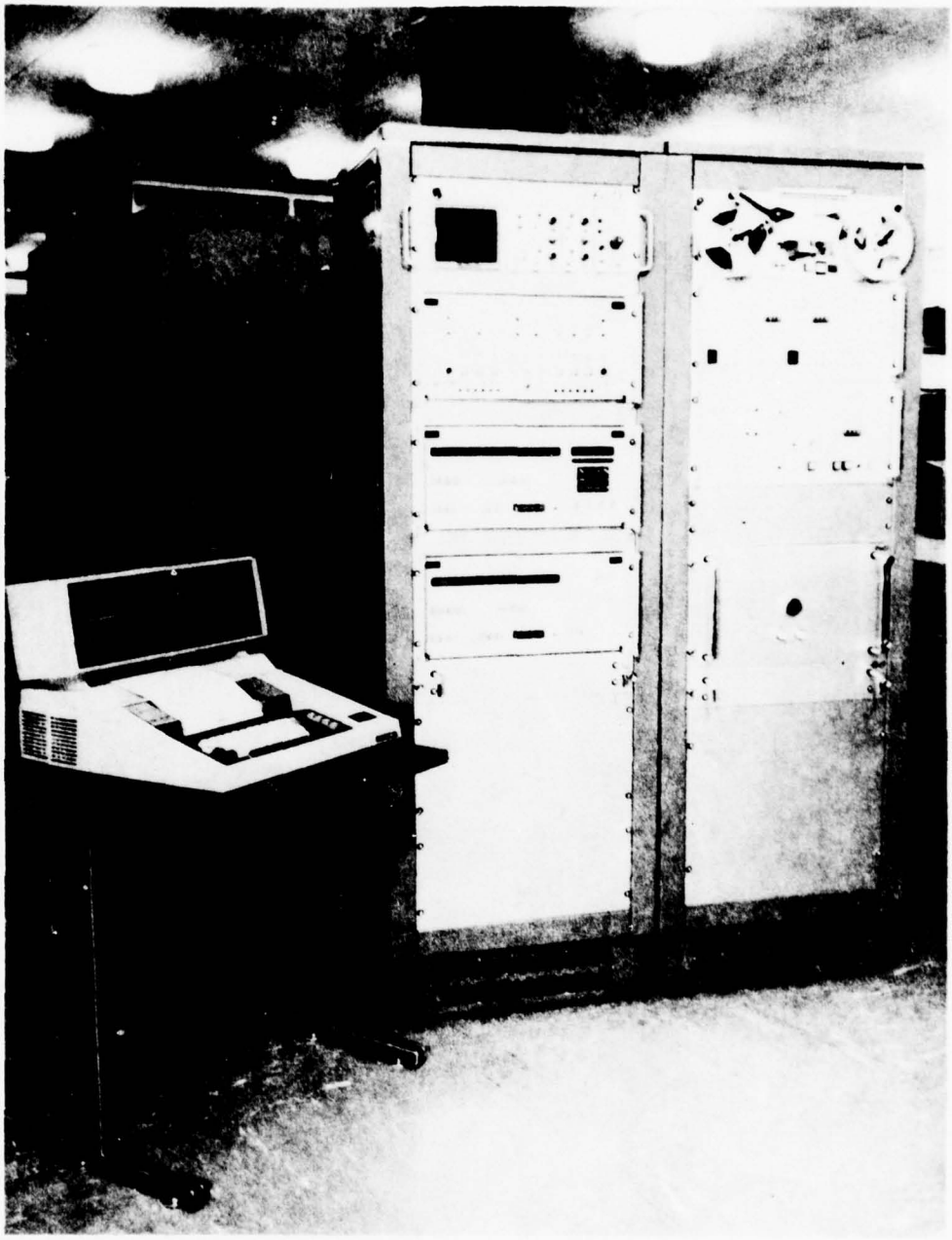


ILLUSTRATION 8. GINA Bus Integration Test Station

# 1553 EXTENSIONS WHICH ALLOW ELIMINATION OF CENTRAL CONTROLLERS

Horst Kister
VDO Luftfahrtgerate Werk, Germany

PRECEDING PAGE BLANK

## 0 Abstract

As an extension to the MIL STD 1553 databus a decentralized control method is described. It can be used in any combination with the standard procedure. A proper adaption of systemarchitecture and control method can give gracefull degradation and easy reconfiguration and therefore bètter system survivebility.

## 1 Introduction

In a contract with German MOD VDO-L developped as an extension to MIL STD 1553 a method for decentralized control of DATA-transfer. The intention was to provide an alternative control-method of DATA-transfer for more sophisticated and distributed systems.

The presented control-method is based on the DATA-channel and wordformat of MIL STD 1553 and it is compatible with this standard and can be used for all the system or only parts of it.

## 2 Allocation of Transmission time

In the basic system the controller is allocating transmission time to the terminals, there is no own activity of the terminal. Now we allow for selfinitiated transmissions of a terminal. For better differentiation in the following these terminals will be called "Member".

With many members being able to initiate transmissions, we have to avoid excess of more than one at a time. For this purpose a time slot-sequence is used. The basic lay-out is shown in fig. 1. All members are listening to the databus. Each "end of transmission" easily is detected. At this instance a counter is reseited and started, using the internal clock, to generate these time-slots. The members are assigned to different time-slots in the sequence of their priorities.

With the priority sequence stepping through, only the member with the highest priority is allowed to start its transmission if it wants to do so. This fact is detected by all other members and causes them to hold back their activities.

At this point normally there is the objection that members at the lower end of priority sequence may not be able to get the bus due to higher priority activities. To avoid this, there has to be spare time in the system. But spare time easily is gained by this control method itself and a alarm procedure will help too. We will discuss this later in more details.

In case of long periods without activity on the bus, synchronization of priority sequence counters in separate members could be lost. To avoid this, a refresh pulse is used. After one frame of the priority sequence without activity, it starts again from the beginning and the first operational member in the sequence generates a refresh pulse. This simulates anend of transmission and therefore resets and synchronizes the priority sequence counters in all members.

To avoid confusion during a random power on-sequence of different members, the priority sequence first is synchronized to the current bus activity and after that the transmitter is enabled.

In reality the priority sequence is a little more complicated, we will come back to it later on.

## 3 Transfer modes

Basically there are only 2 transfer modes: First there is the

### 3.1 Transmission mode

Transmission mode is initiated by a member itself and always starts in the time-slot, only this member is assigned to (see fig 2). It can be

- periodically with a fixed transfer rate
- periodically with variable transfer rate for instance dependent of change rate of this particular parameter or
- a randomly generated transfer.

In the

3.2 Request-reply mode

A member of the system can "ask" for a particular information by issuing a "request." The adressed member can respond immediately with a "reply message". A request is initiated like a "transmission". The requested member starts the "reply message" in a special timeslot R (shown in fig. 3) before any other activities. So a reply message normally follows immediately to the request.

If the requested information is not available and has to be generated either by conversion or calculation, the reply message only contains the information "more time needed". After preparation of data, the requested member then starts a standard transmission.

## 4. Message structure

### 4.1 Transmission message

A transmission message as shown in fig. 4, uses the broadcast-method. Therefore it carries the source address in the command word, we call it a 'tag'-command instead of receive-command. It is up to the listening members to pick up those information they are programmed to. This will be explained in more detail in a later paragraph.

### 4.2 Request-reply message

For the request a standard transmit-command is used (fig. 5). The reply is a transmission as defined before.

## 5 Alarm mode

Introduction of the time-slot-sequence makes it possible to add a really fast alarm mode (see fig. 6).

For this purpose a member can issue a short pulse in the time slot No. 13 . indicating, that it wants to issue a time critical message. All other members listening to the bus now will hold back.their normal activities. In the adjacent priority sequence only the alarm message is transmitted.

If more than one member issues an alarm-pulse at a time, the higher priority transmits its message first, then the next one repeats the alarm-pulse and will transmit in the adjacent priority-sequence and so on.

The superposition of alarm-pulses may cause a distortion due to propagation delays on the bus. But there is no difficulty for the receiver circuit to detect it properly.

So with the alarm-pulse the system switches to a higher priority level and falls back after all alarm messages are transmitted.

If accidentally a noise pulse appears in the alarm slot the adjacent sequence will run through without activity, but the system then falls back to normal operation.

## 6 Supervisor

Status monitoring and testing of a system basically is a central function. Since there is no central controller, the system can be monitored by a "Supervisor". Its main duty is to listen to the communications and to compare parameter refresh rates with predetermined limits. If there normally is no periodic emission, the supervisor can "ask" the member using the request-reply mode. A statusinformation with failure indication as well as a missing reply leeds to a failure warning.

Because the supervisor is limited to monitoring functions its breakdown will not effect busoperation.

## 7 Safety circuit

### 7.1 Chatterswitch

The problem of unallowed emissions of a member due to hard- or software failure is common all databusses centrally or decentrally controlled. To minimize this "chattering" problem, we designed a failsafe circuit limiting the average of transmission time. The principle is shown in fig. 7. The transmission energy is taken from a capacitor. It is charged through a series resistor.

The added circuit stabilizes the output voltage and switches it off if allowed average is exceeded. This circuit has not to be failsafe, it's for comfort only.

### 7.2 Interference detector

Due to severe noise or hardware problems, more than one terminal or member may take the bus at a time. To avoid more than necessary confusion of the system, an interference detector is added. The principle is shown in fig. 8. It compares signals on the bus with its own output. So signals from other members easily are recognized and the current transmission is shut down. So the priority sequence starts and the system tries again.

## 8 Compatibility with MIL STD 1553

We are well aware that it may be necessary to use also standard equipment of the shelf using the basic MIL STD procedure.

In a mixed system (fig. 9) for a terminal to member transfer these terminals will be initiated by members issuing a transmit-command. They will reply in accordance with the standard in the 2...10 $\mu$s gap (see fig. 10).

A terminal to terminal transfer needs the help of another member playing the controllers roll. The resulting communication is identical to the standard and is shown in fig. 11. The special time-slot C can be used for this controller giving it highest priority (fig. 12).

9 Description of experimental hardware (fig. 13)

Using standard MSI circuits, a "databus module" was developed. For experimental work 5 subsystem-simulator boxes where used to check out all the functions described before.

The blockdiagram of the databus-module is given in fig. 13 .

The interface to the host-system is designed for easy adaption to all parallel or serial bus oriented structures.

The basic blockdiagram should be selfexplaining. So I can concentrate on some details.

First of all there is a pin to select the MIL STD procedure or the VDO-procedure.

On the receiving side we added a ROM. Its addresslines are controlled by all address bits and the T/R bit of the received commandword.

In the broadcast-mode it allows to program all source addresses and subaddresses this member shall pick up. So only those information will cause a "data ready" and a interrupt to the subsystem.

This ROM also contains the members address. If its own address is detected in a "request" "data ready" interrupts the host-system for proper reply-message.

If the host-system is not able to respond in the available time of 18 µs, a preloaded additional shiftregister will automatically issue a "wait" message.

After requested data are available they are transferred via a "transmission".

The additional shiftregister also can be used to meet the 1o us deadline for statusinformation in the MIL STD procedure.

And last not least there is the priority-center generating the time-slots. This is the only additional hardware for the decentral control procedure and represents about 4 % of the total amount of hardware. In a future LSI circuit it can be neglected.

1o Influence to system design

The easiest approach is to program the members for periodic or not periodic transmissions of its various parameters. At the receiving side it may be programmed to pick up what it is interested in, and, if there is no sufficient information to "ask" for it. And then forget all worries, the system will take care of itself.

I know, that this is a systems-engeneer's nightmare. He is used to know exactly what is going on in the system in respect to time and sequence.

So let's do very little steps.

- First there is the small system (like fig. 14) using a central computer controlling some pheripheral elements and receiving information from them, if the central computer fails, all the system fails. In this case there is no senseful application of the decentral bus control.

- Second there is basically the same system, but it is maybe senseful, to preserve some communication between peripherals as a standby function in case of a computer failure (see fig. 15). If operational, the computer controls the bus using a high priority time-slot. If it fails, the peripherals will continue to operate, using their lower priority time-slots.

- Third, there is the system where some of the terminals have a random and seldom information output (fig. 16). In the basic system they have to be scanned with the worst case rate. In the decentral approach they will take the bus only if necessary. And if urgent, they can use the alarm mode.

- Fourth, there is the system with functional independent special to purpose subsystems, but using partial the same data source and sinks (fig. 17). If there are time sensitive data, for instance if a differentiation takes place, each subsystem can control the relevant sinks and sources using a fixed rate and a given priority. If one subsystem fails, the other will not be effected.

What I wanted to show is, that there is not only the central or the decentral control, but any combination in between.

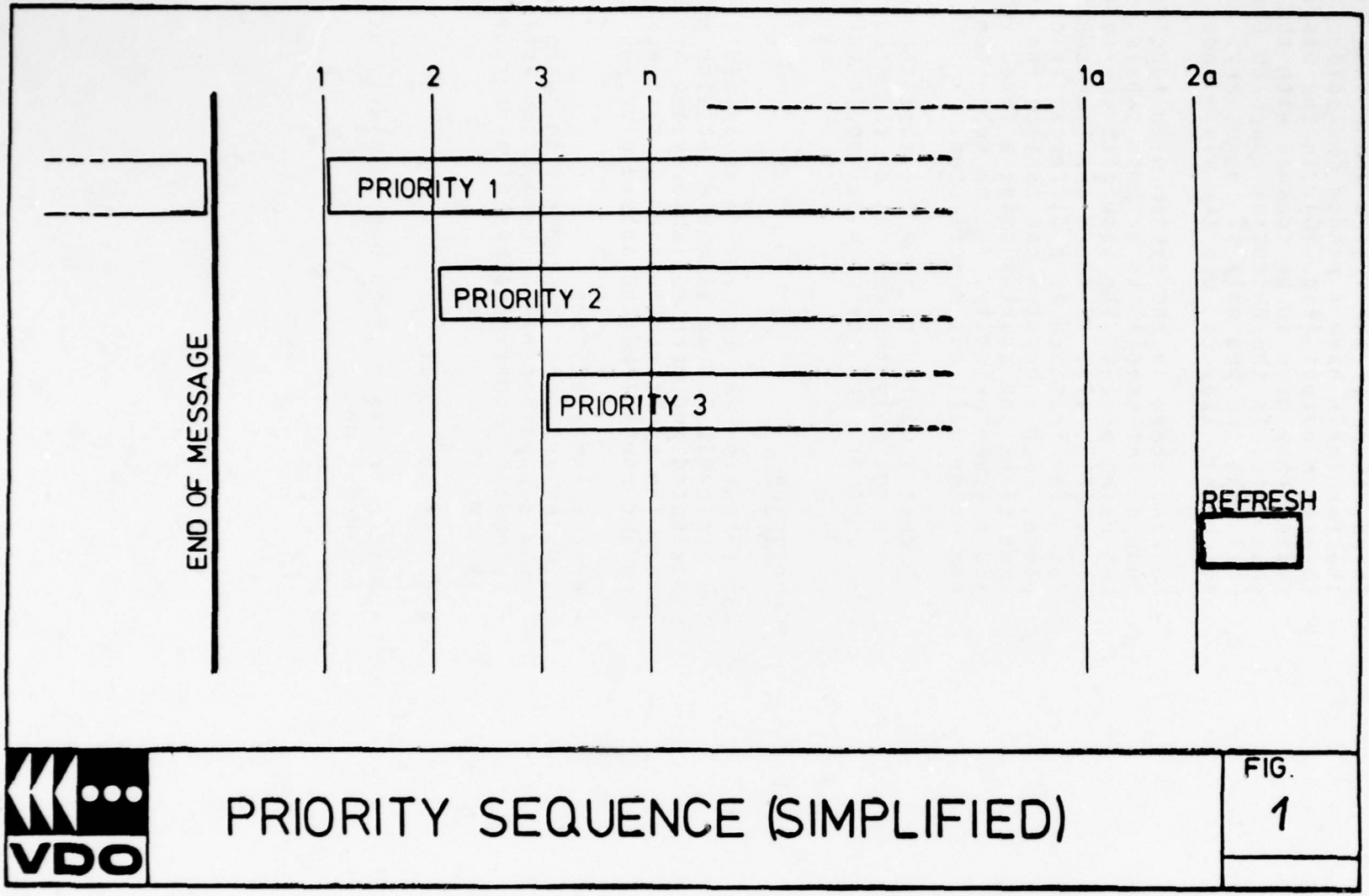
## 11 Conclusion

Our intention was to provide as an addition to the basic MIL STD an adequate tool for more distributed and more reliable system architectures. The right combination of systemarchitecture and control method should give you

- graceful degradation,
- simpler and more transparent software due to good separation of autonomous subsystems,
- automatic reconfiguration of a degraded system

and last not least

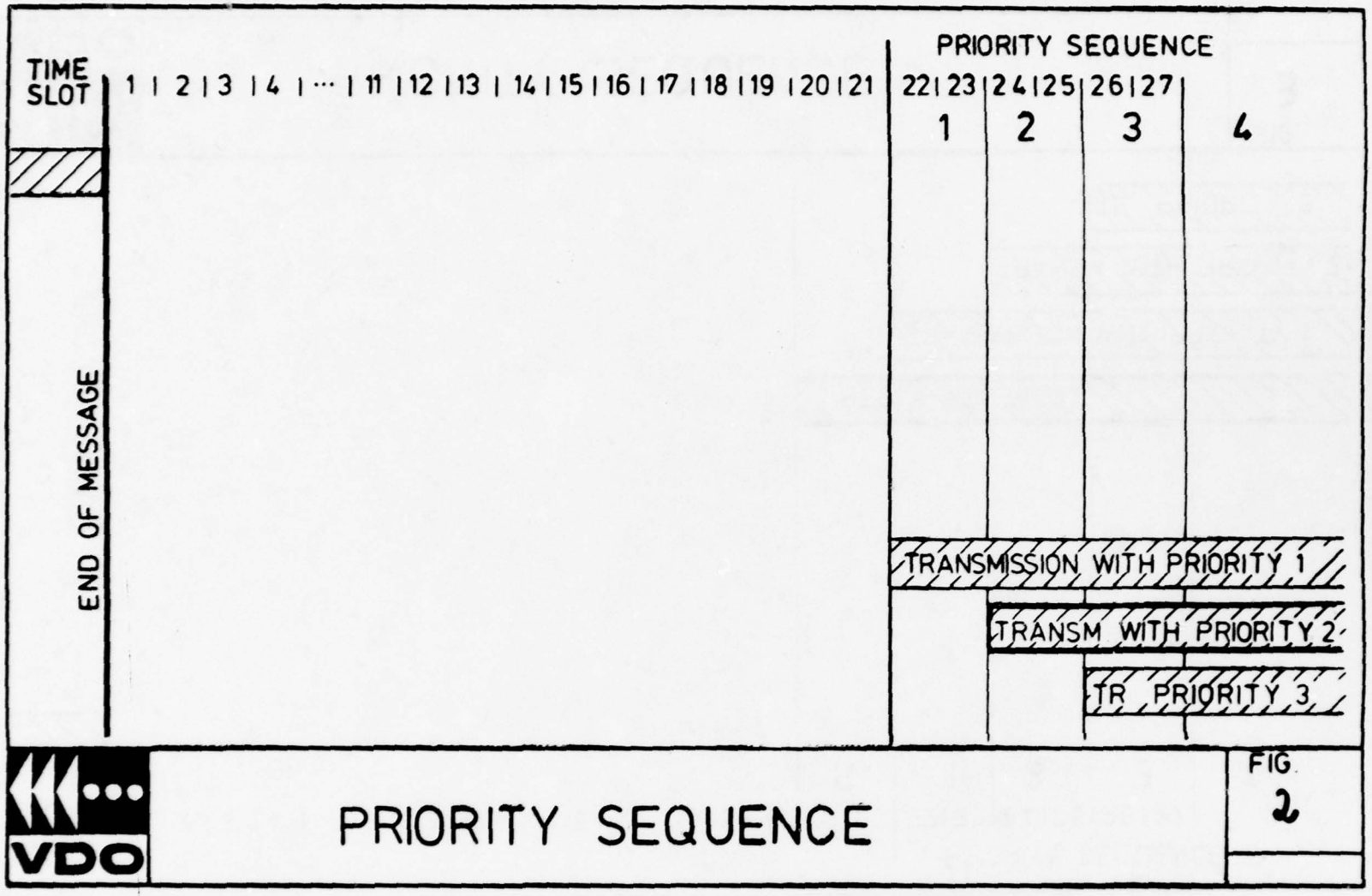
- better transfer power due to widely usable broadcast mode.

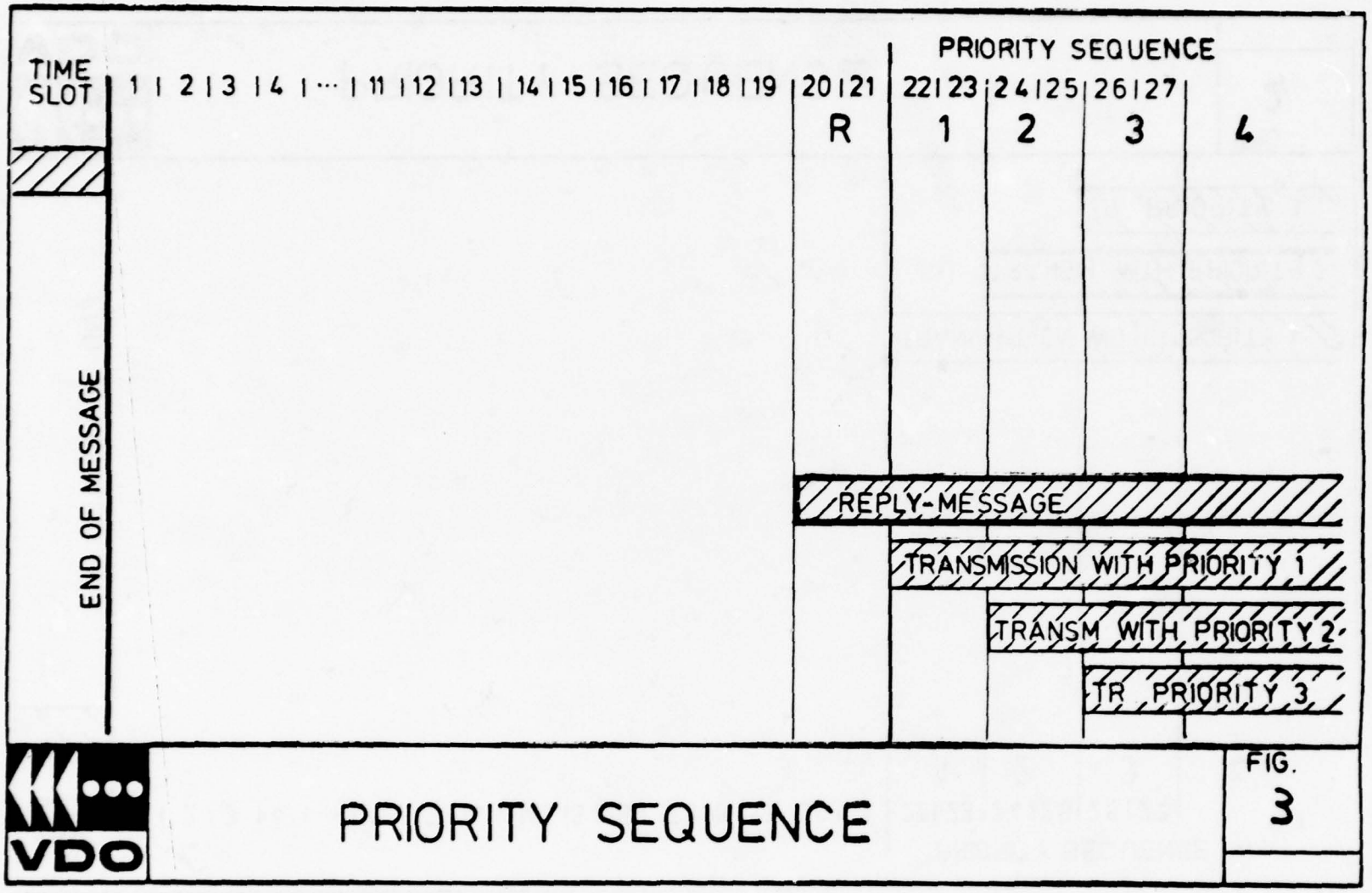


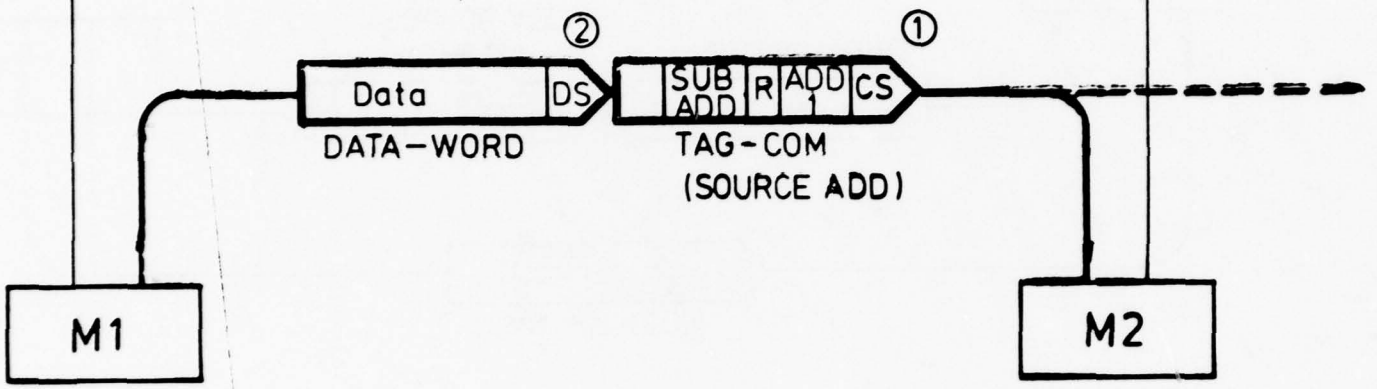

VDO

PRIORITY SEQUENCE (SIMPLIFIED)

FIG. 1


TIME SLOT
1 | 2 | 3 | 4 | ··· | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21
END OF MESSAGE
PRIORITY SEQUENCE
22 | 23 | 24 | 25 | 26 | 27
1
2
3
4
TRANSMISSION WITH PRIORITY 1
TRANSM WITH PRIORITY 2
TR PRIORITY 3
VDO
PRIORITY SEQUENCE
FIG
2

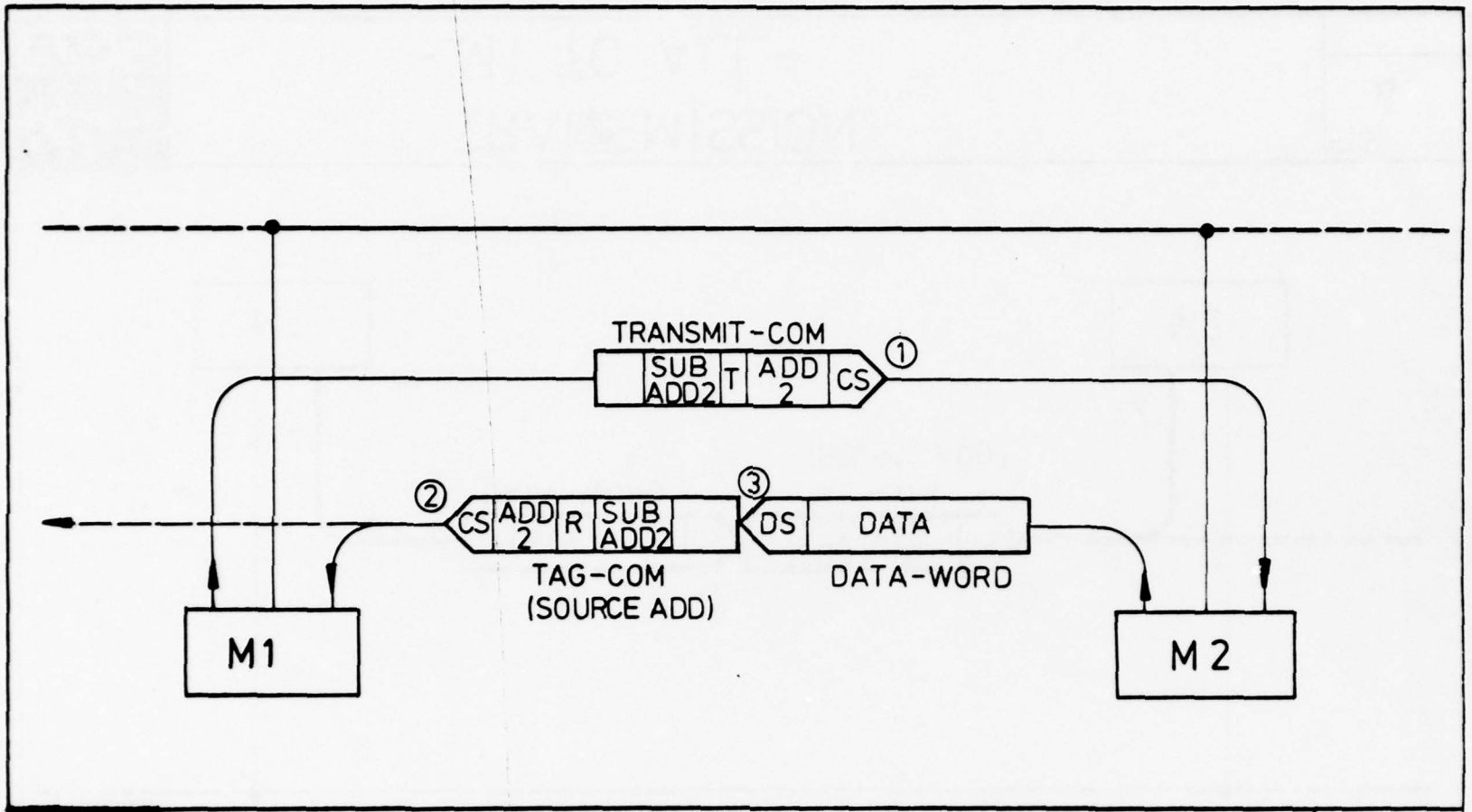

TIME SLOT
1 | 2 | 3 | 4 | ··· | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21
PRIORITY SEQUENCE
22 | 23 | 24 | 25 | 26 | 27
R
1
2
3
4
END OF MESSAGE
REPLY-MESSAGE
TRANSMISSION WITH PRIORITY 1
TRANSM WITH PRIORITY 2
TR PRIORITY 3
VDO
PRIORITY SEQUENCE
FIG. 3




VDO

# TRANSMISSION
## - M1 TO ALL -

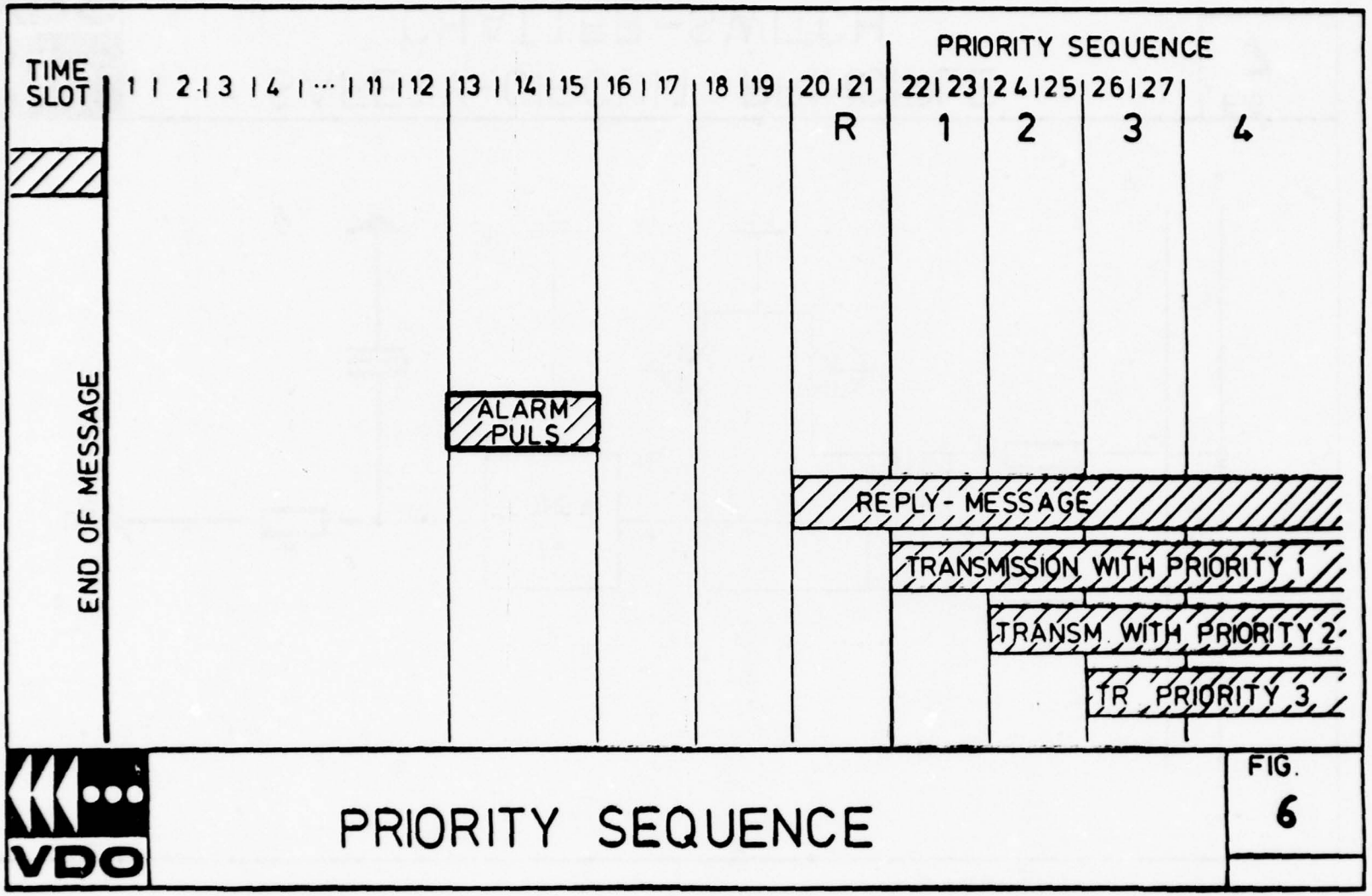
FIG. 4

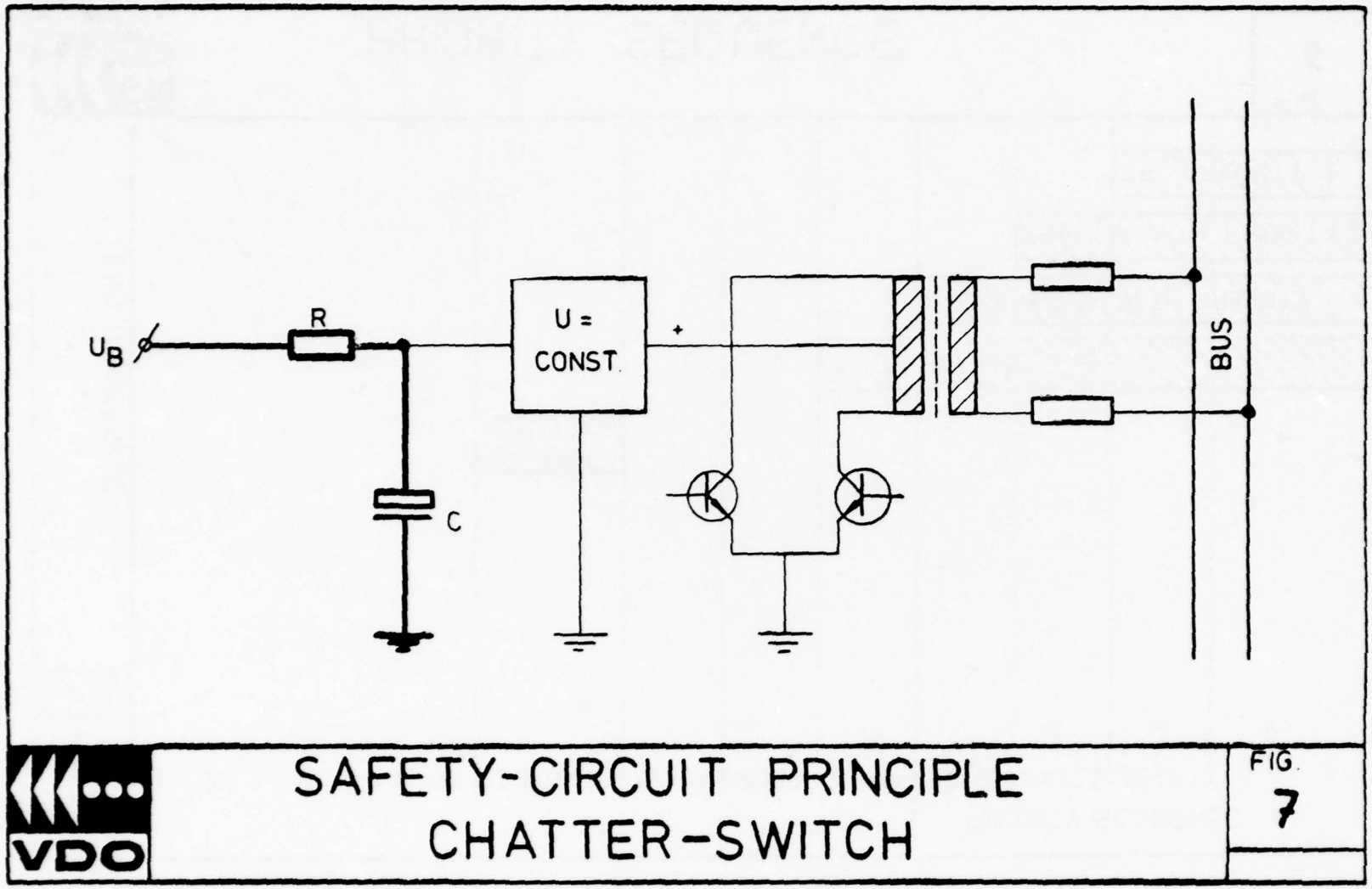



VDO

# REQUEST / REPLY
## - M1 REQUESTS M2, M2 REPLIES TO ALL

FIG. 5

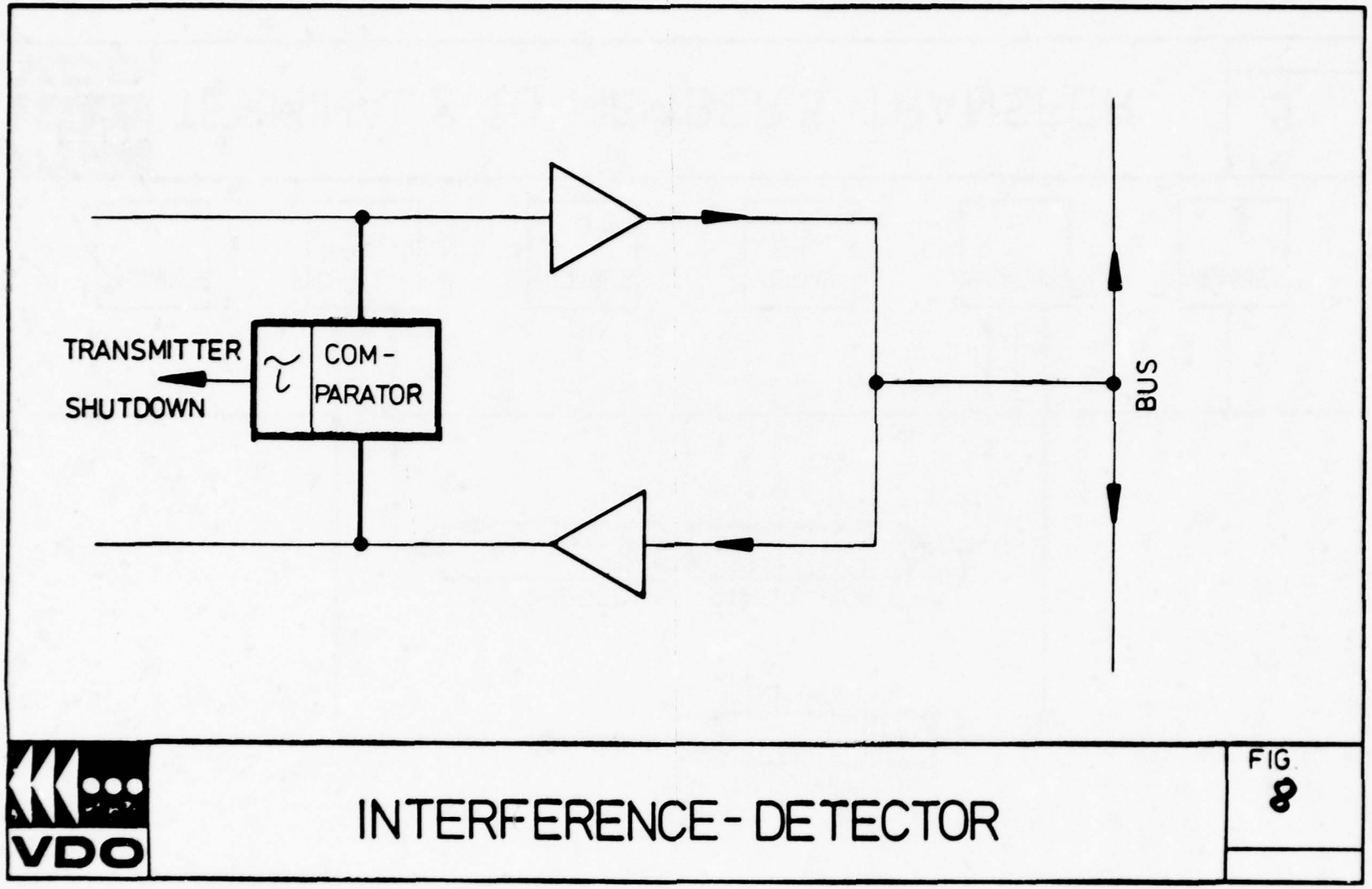

TIME SLOT
1 | 2 | 3 | 4 | ··· | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27
PRIORITY SEQUENCE
R
1
2
3
4
END OF MESSAGE
ALARM PULS
REPLY-MESSAGE
TRANSMISSION WITH PRIORITY 1
TRANSM WITH PRIORITY 2
TR PRIORITY 3
VDO
PRIORITY SEQUENCE
FIG. 6

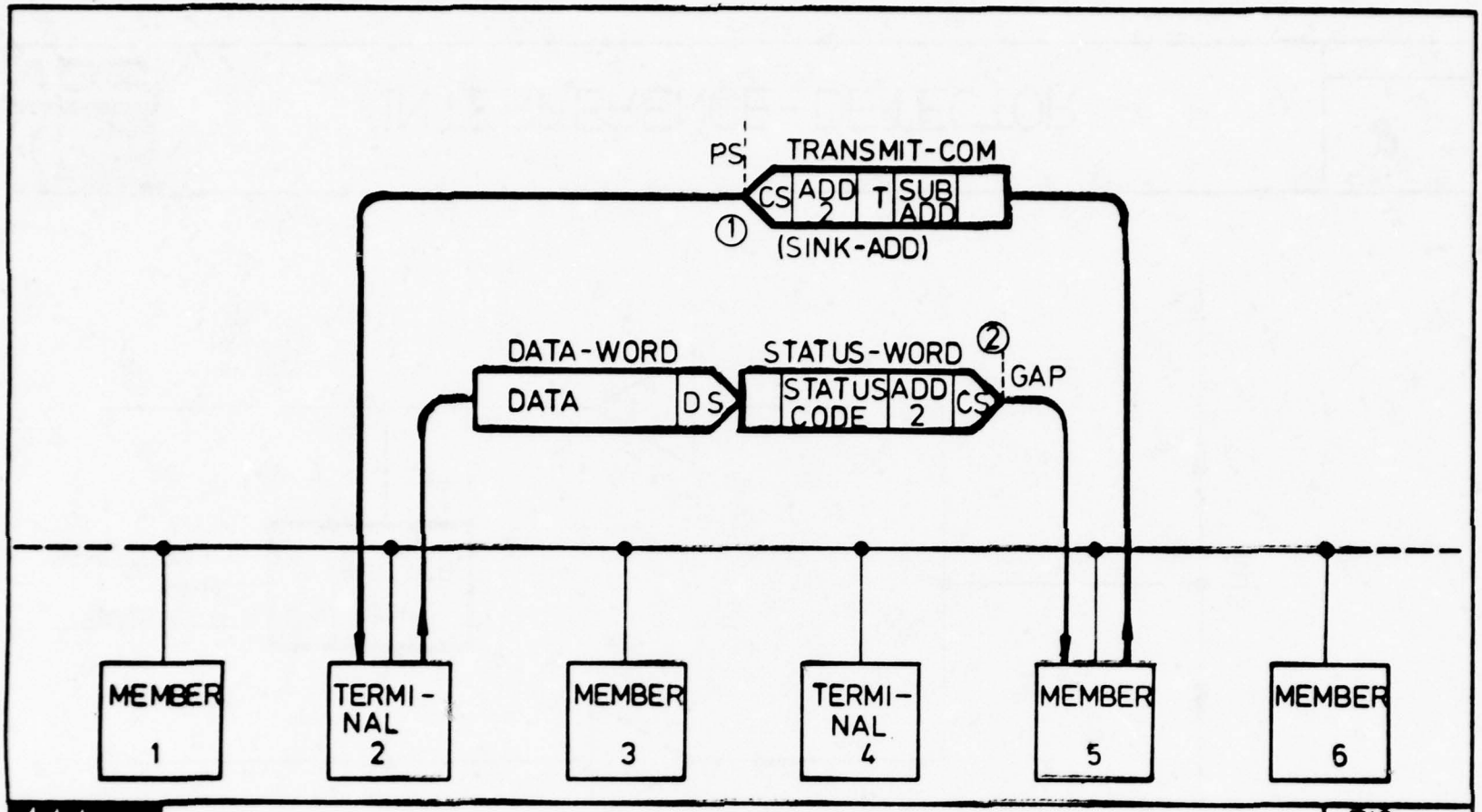



VDO

# SAFETY-CIRCUIT PRINCIPLE CHATTER-SWITCH

FIG. 7

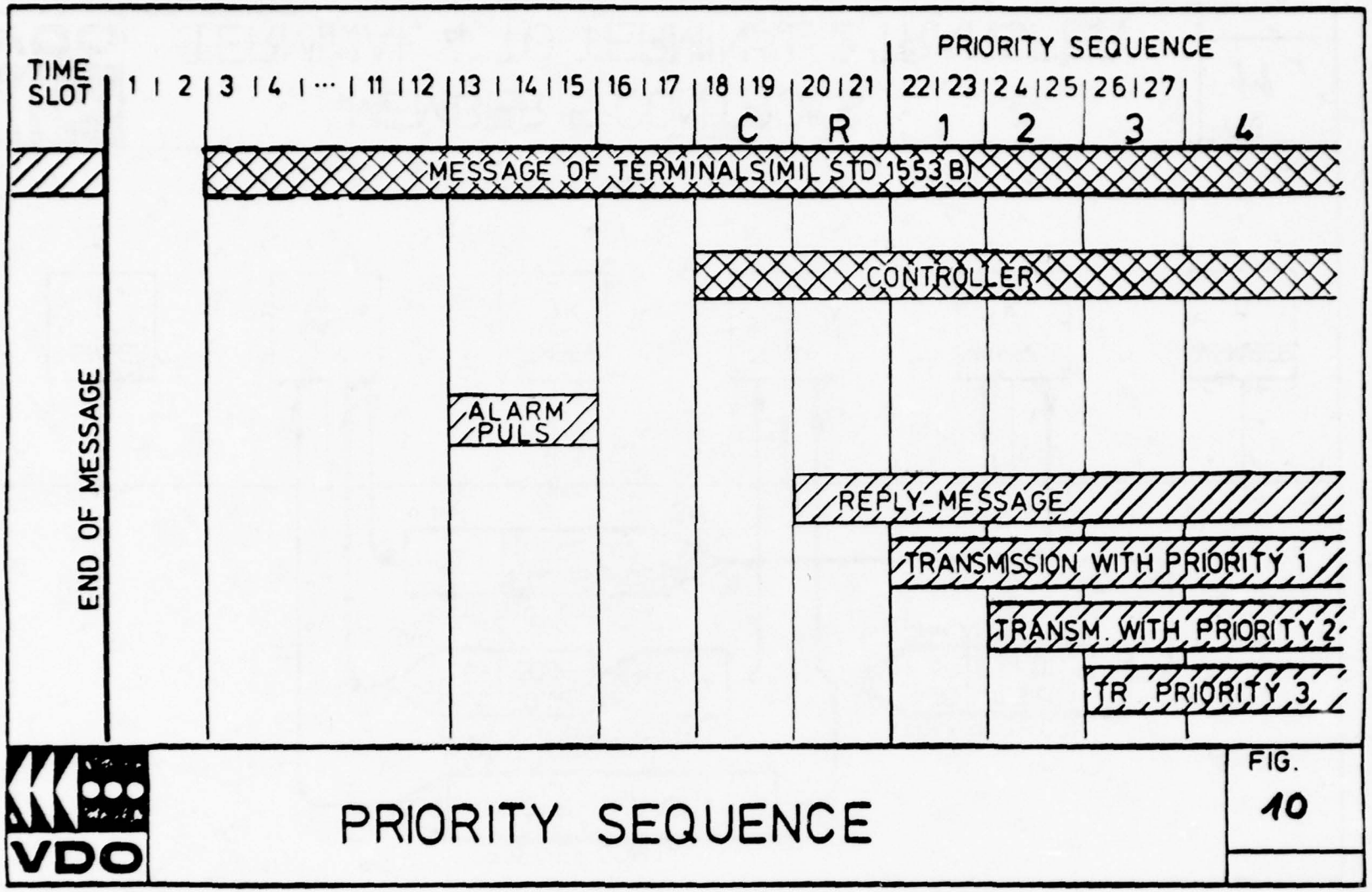

TRANSMITTER
SHUTDOWN
COM-
PARATOR
BUS
VDO
INTERFERENCE-DETECTOR
FIG
8

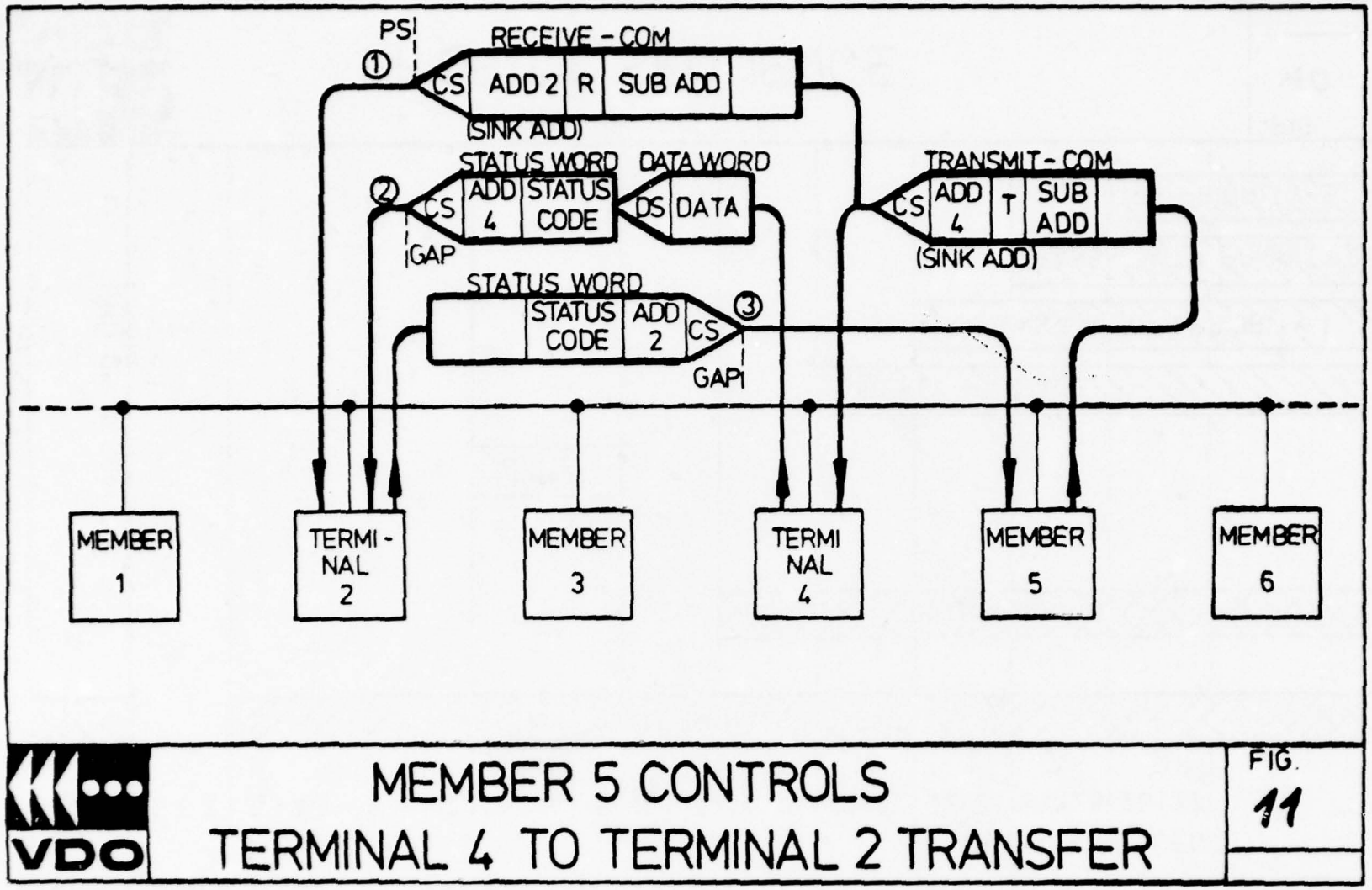



VDO

# TERMINAL 2 TO MEMBER 5 TRANSFER

FIG. 9

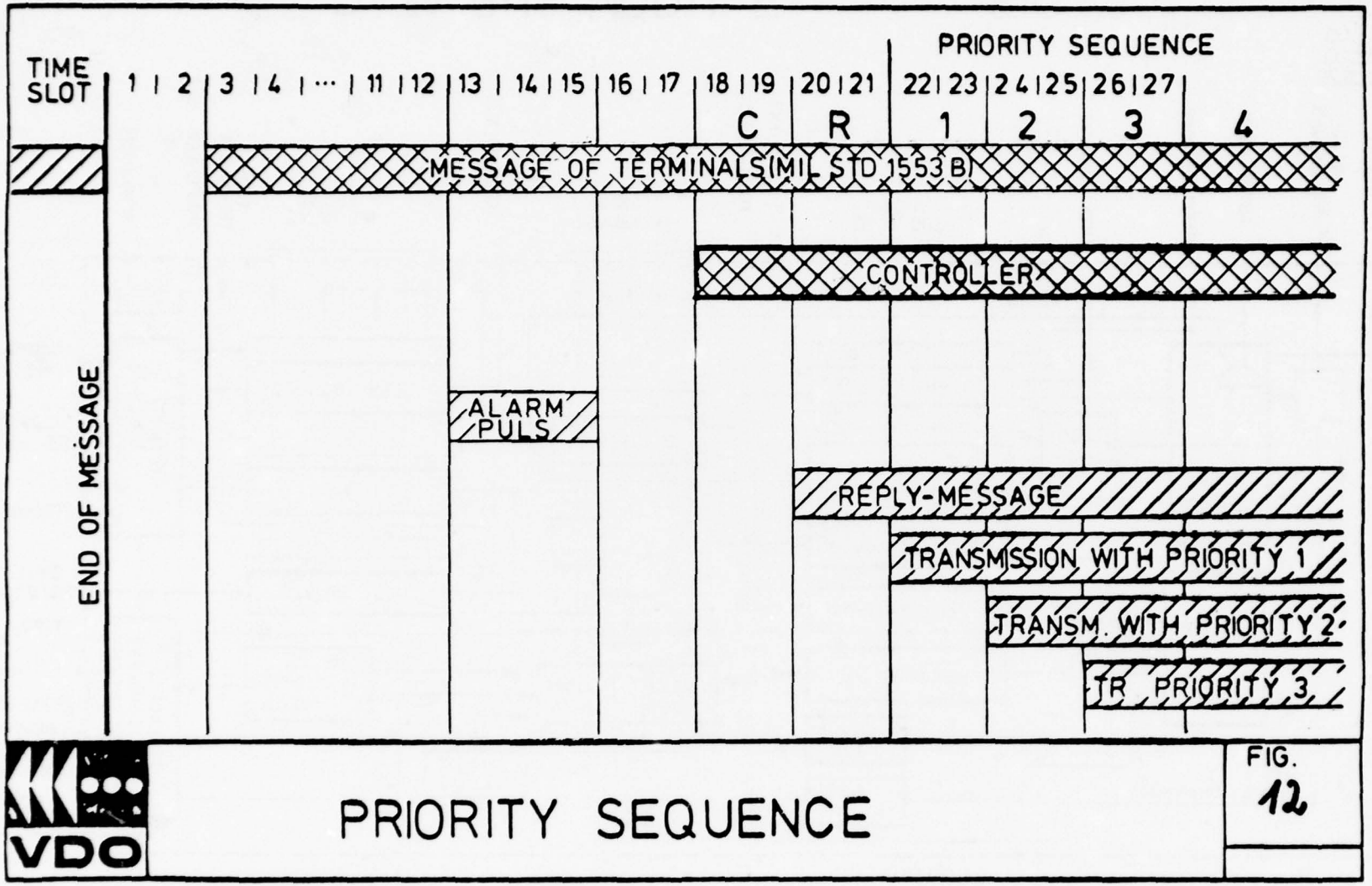

TIME SLOT
1 | 2 | 3 | 4 | ··· | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27
PRIORITY SEQUENCE
C
R
1
2
3
4
MESSAGE OF TERMINALS(MIL STD 1553 B)
CONTROLLER
ALARM PULS
REPLY-MESSAGE
TRANSMISSION WITH PRIORITY 1
TRANSM. WITH PRIORITY 2
TR PRIORITY 3
END OF MESSAGE
VDO
PRIORITY SEQUENCE
FIG. 10

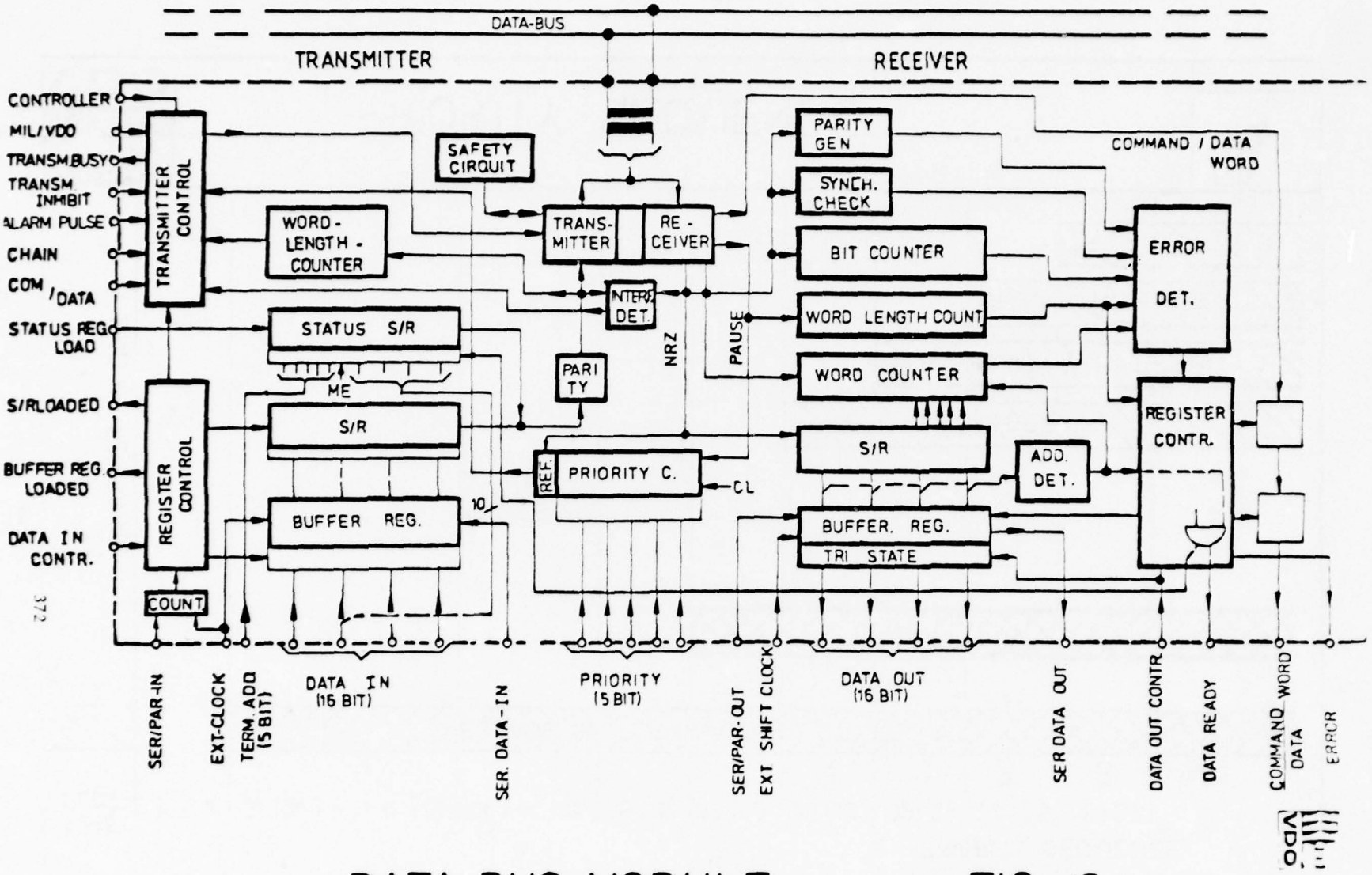



VDO

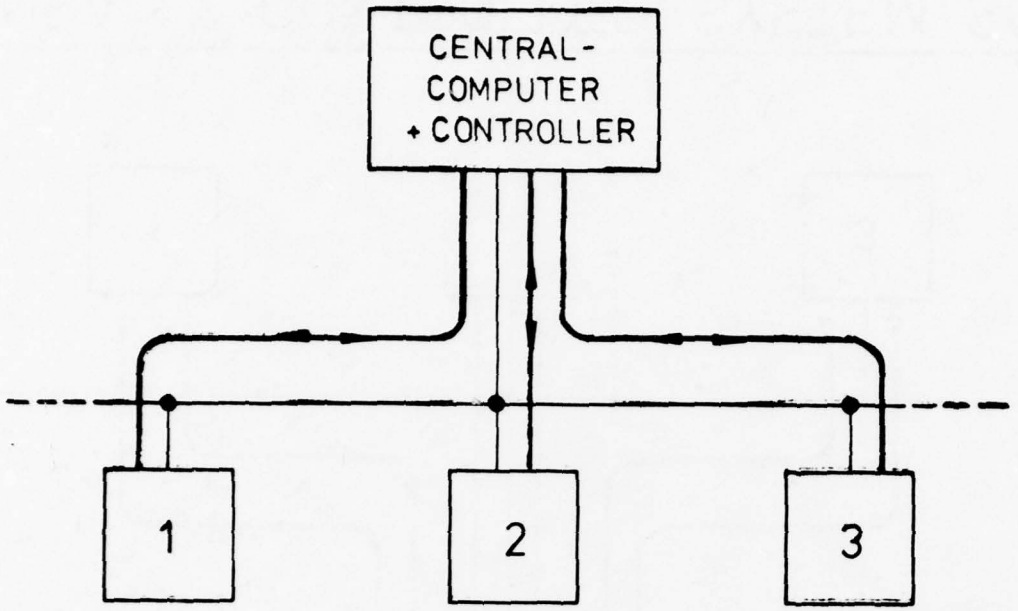
# MEMBER 5 CONTROLS
# TERMINAL 4 TO TERMINAL 2 TRANSFER

FIG. 11


TIME SLOT
1 | 2 | 3 | 4 | ··· | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27
PRIORITY SEQUENCE
C
R
1
2
3
4
MESSAGE OF TERMINALS (MIL STD 1553 B)
CONTROLLER
ALARM PULS
REPLY-MESSAGE
TRANSMISSION WITH PRIORITY 1
TRANSM. WITH PRIORITY 2
TR PRIORITY 3
END OF MESSAGE
VDO
PRIORITY SEQUENCE
FIG. 12



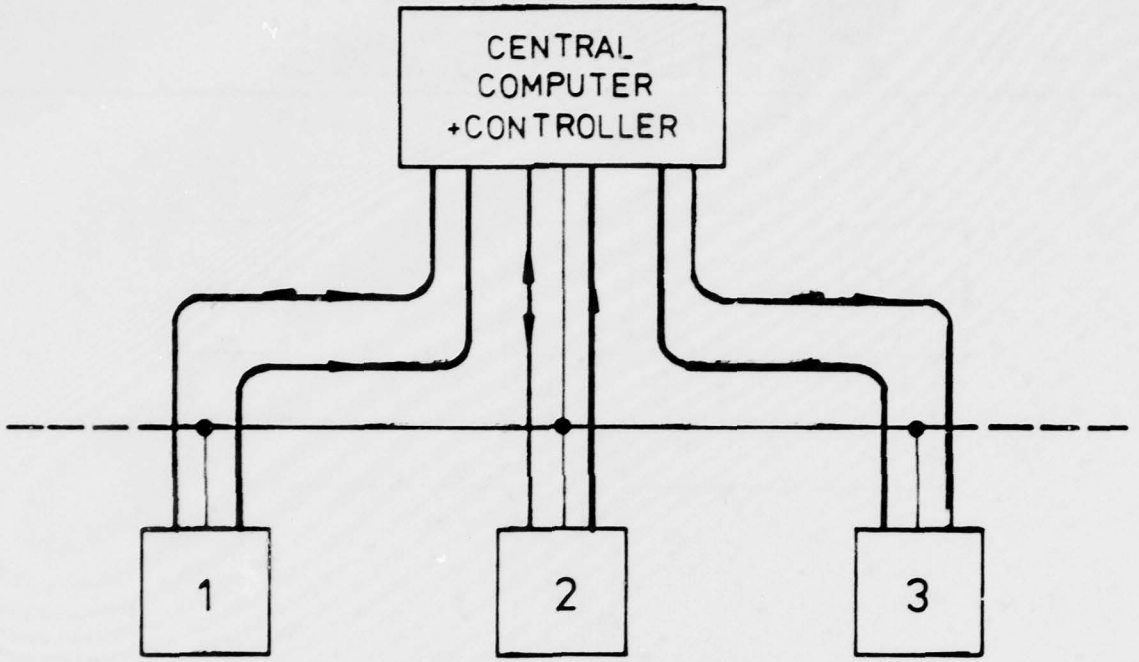

DATA BUS MODULE FIG. 13




VDO

SIMPLE CENTRALLY CONTROLLED SYSTEM

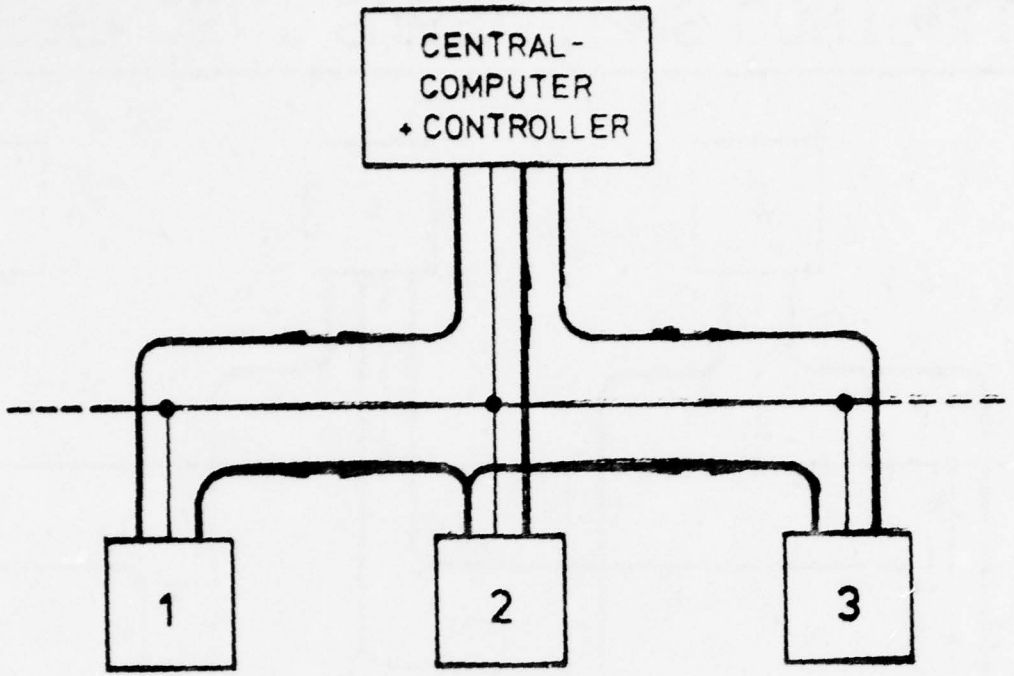
FIG. 14




VDO

CENRALLY CONTROLLED SYSTEM, SOME PERIPHERALs WITH RANDOM OUTPUT

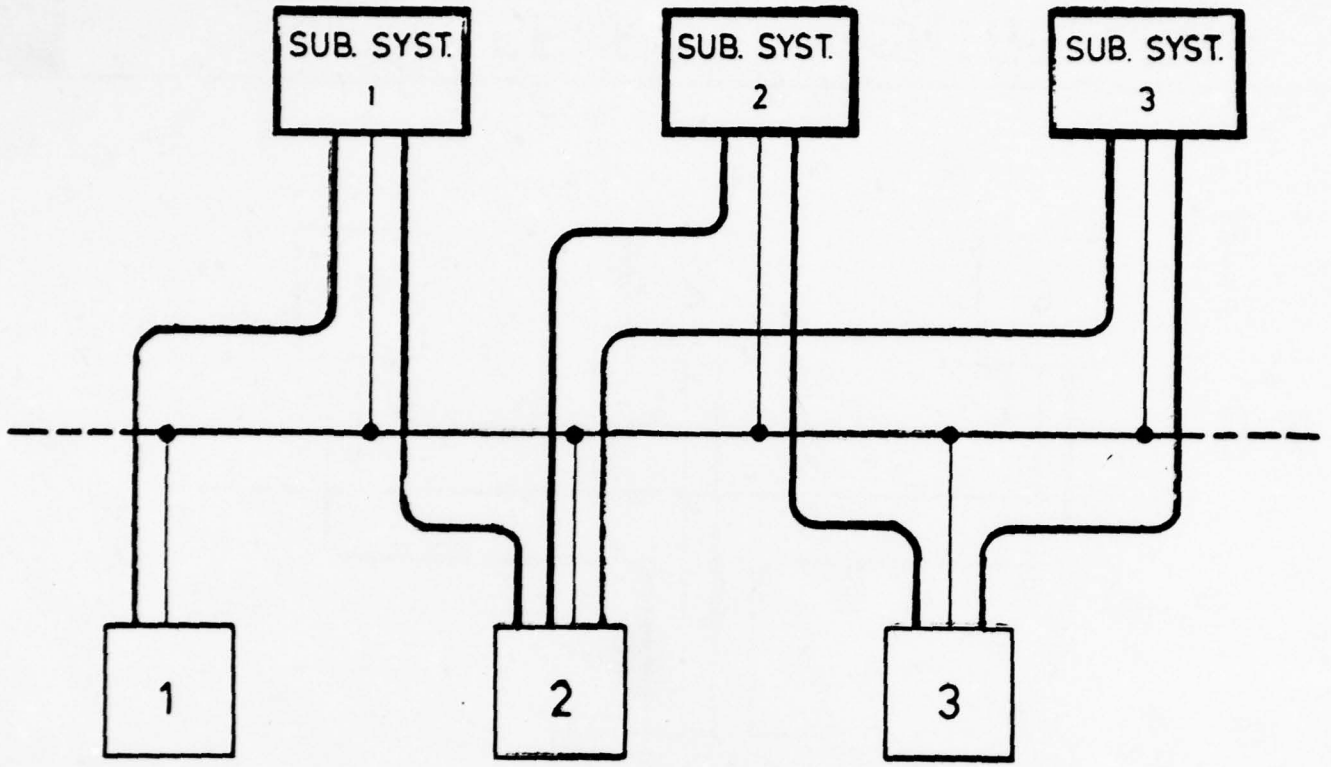
FIG 15




VDO

SIMPLE CENRALLY CONTROLLED SYSTEM WITH BACK-UP MODE

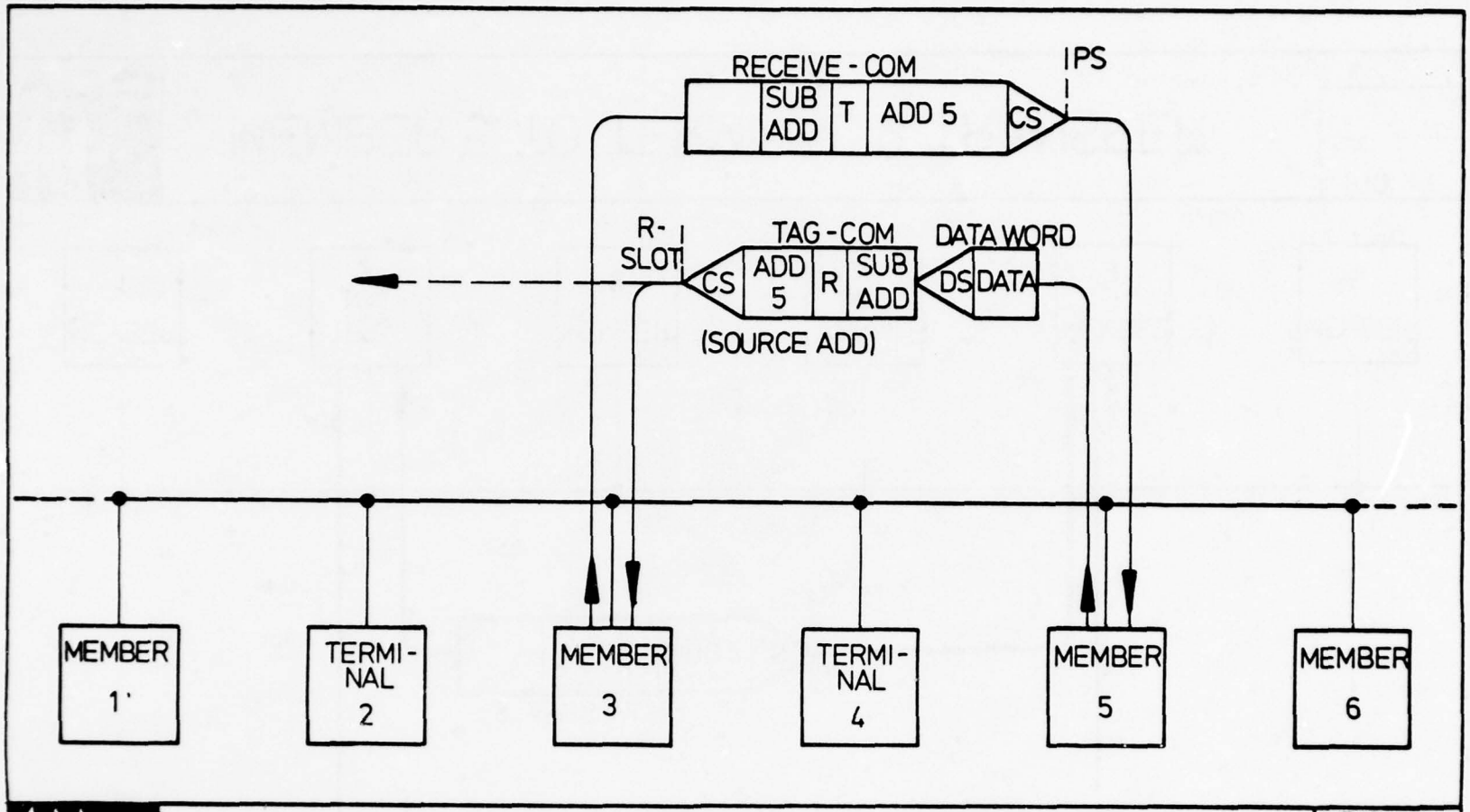
FIG. 16




VDO

INDEPENDENT SUBSYSTEMS CONTROLLING SAME SINK AND SOURCES

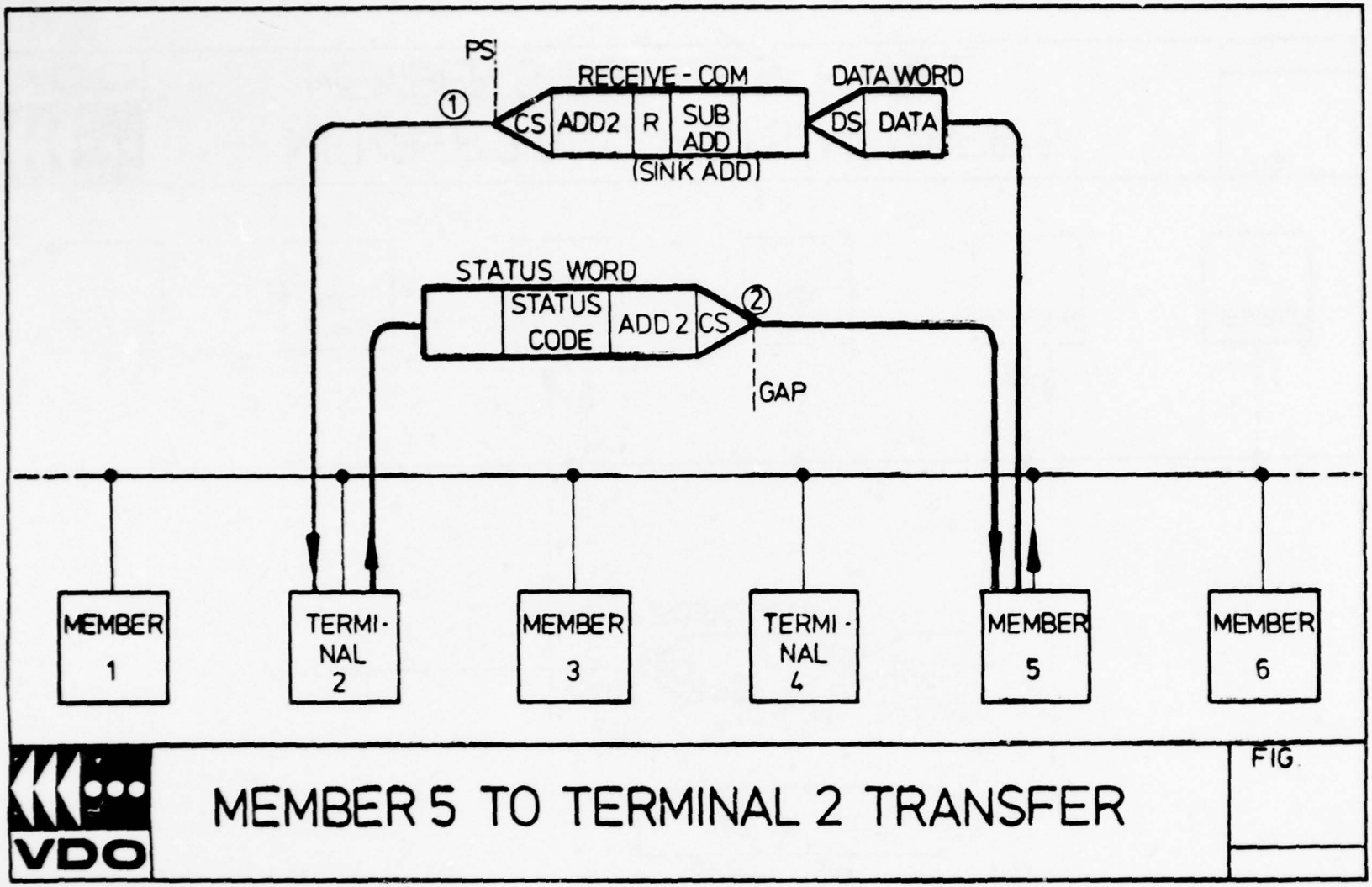
FIG. 17




VDO

MEMBER 3 REQUESTS MEMBER 5
MEMBER 5 REPLIES TO ALL

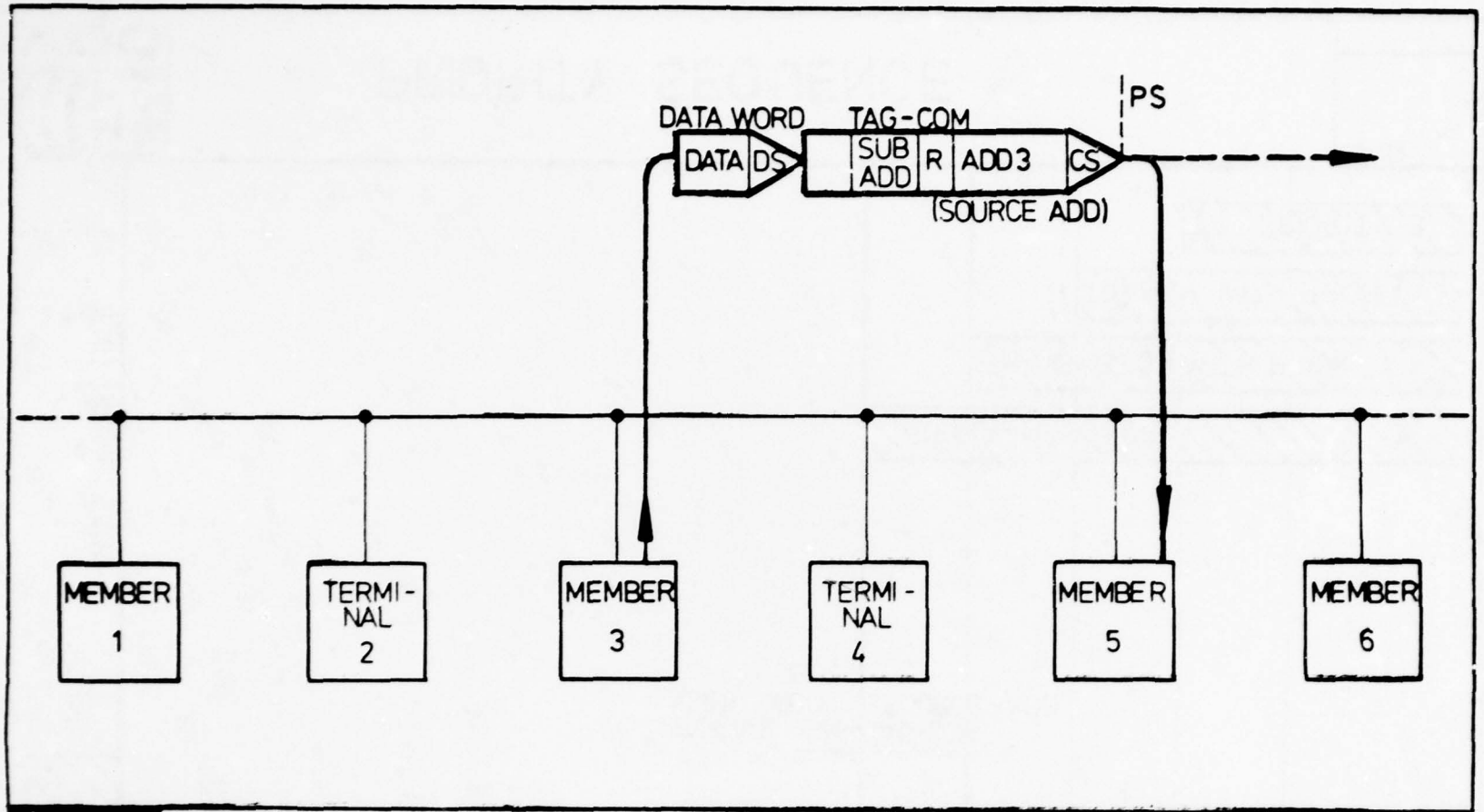
FIG.




VDO

MEMBER 5 TO TERMINAL 2 TRANSFER

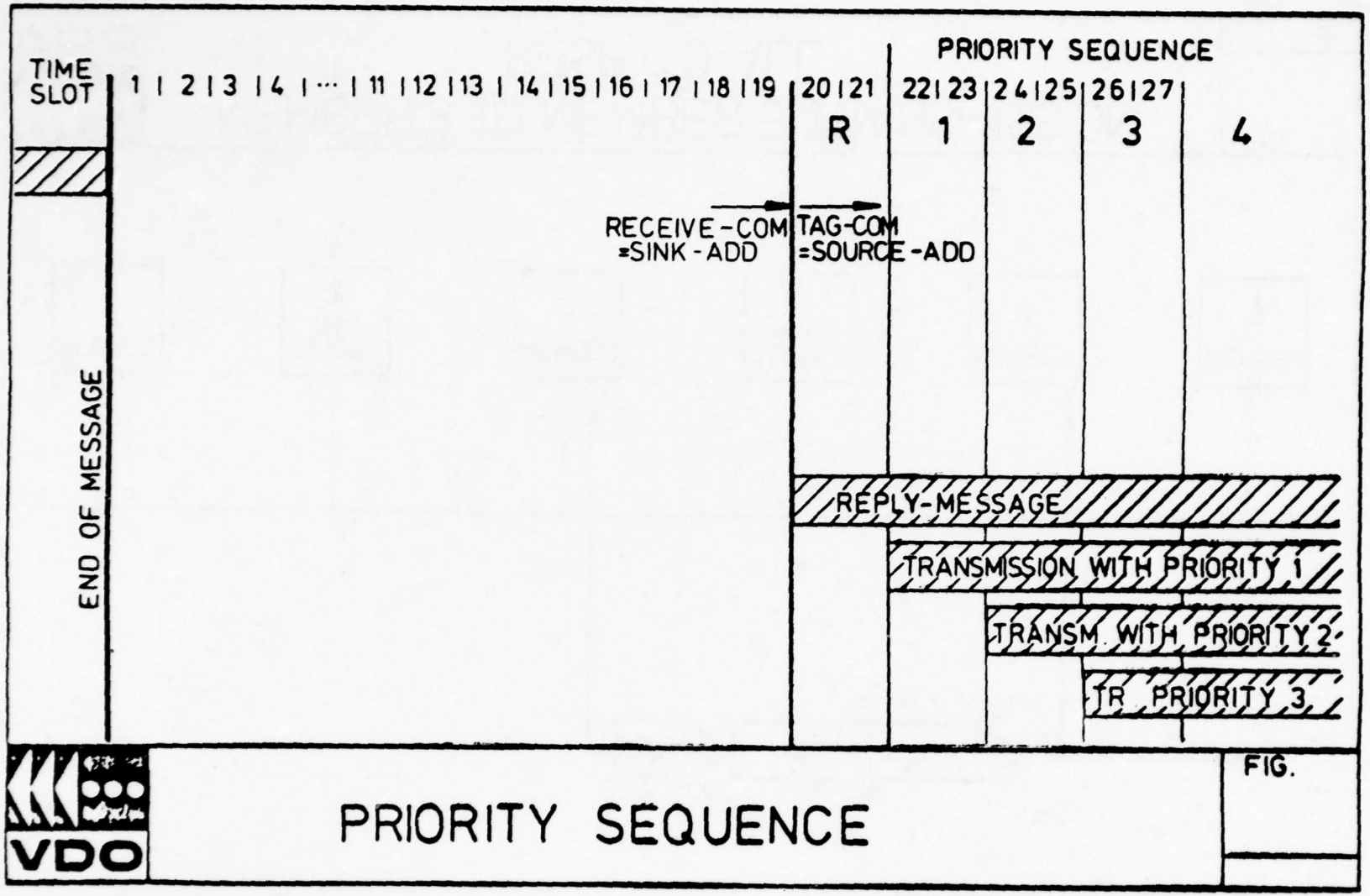
FIG




VDO

# MEMBER 3 TO MEMBER 5 TRANSMISSION RESP. TO ALL

FIG.


TIME SLOT
1 | 2 | 3 | 4 | ··· | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27
PRIORITY SEQUENCE
R
1
2
3
4
RECEIVE-COM =SINK-ADD
TAG-COM =SOURCE-ADD
END OF MESSAGE
REPLY-MESSAGE
TRANSMISSION WITH PRIORITY 1
TRANSM. WITH PRIORITY 2
TR PRIORITY 3
VDO
PRIORITY SEQUENCE
FIG.

# USE OF DIGITAL DATA BUSES ON COMMERCIAL AIRCRAFT

V. J. Small

Boeing Commercial Airplane Company

## ABSTRACT

The next generation of commercial aircraft will make extensive use of digital data buses. The commercial aircraft digital data buses differ from the MIL-STD-1553 buses in concept and operation. Commercial buses have developed by an evolutionary process, to meet the requirements of each generation of commercial aircraft. Two digital data buses have been defined for the next generation of commercial aircraft; ARINC 429 and ARINC 453. These buses will meet all the bus requirements of the next generation of commercial aircraft which will utilize a largely digital avionics suite. The two standards are capable of growth to meet possible new requirements of future generations of commercial aircraft.

## INTRODUCTION

The present generation commercial aircraft utilize digital data buses on an individual system basis, i.e. Inertial Navigation System, digital air data, etc. The next generation of commercial aircraft will make extensive use of data buses on an aircraft wide, architectural basis.

To describe the use of digital data buses on the next generation commercial aircraft, it is instructive to review the historical development of digital data transfer systems by the commercial aircraft community, and the role played by Aeronautical Radio Inc., (ARINC). ARINC is supported by, and is responsible to, the airline industry. ARINC sponsors the Airline Electronic Engineering Committee (AEEC) which formulates standards in the form of characteristics and specifications for airline electronic equipment and systems. AEEC subcommittees provide an open forum where avionic system development and characteristic preparation can be

conducted with all interested parties present.

With the advent of digital technology, serial digital data buses evolved rather naturally to replace their analog counterparts. The serial data buses were developed by the groups developing the basic sensors. Each sensor had several parameters to be transmitted on the bus for reception by one or more user terminals.

A word format was developed which consisted of a label to indicate the particular parameter being sent, followed by the parameter data and any sign or status indications. The sensors and indicator units could be built and supplied by different manufacturers, therefore, the data transmission system was fully defined to ensure unit interchangeability. This development philosophy led to several features which were first incorporated in 1971 into the "digital air data system" characteristic ARINC 575.

- o A broadcast bus satisfied the need for sending labeled parameters from a sensor to possibly several indicators.
- o The bus was transmitter controlled, i.e. the transmitter sends a data word when the word is available to be sent.
- o As in analog systems, the indicator was considered a data sink.
- o A complete data message was contained in one word.
- o Each word had the parameter type identified with a label.
- o The transmission system was fully defined to permit unit interchangeability on the bus and interchangeability of units between airplane types.
- o The bus used a single twisted shielded pair wire and employed RZ bipolar modulation of the data.
- o An electro-mechanical indicator could not respond to rapid changes in input digital signal, therefore, an error control policy of rapid refresh of information was adopted.

AEEC at this time recognized the potential for uncontrolled proliferation of data standards all loosely based upon the ARINC 575 system. In 1973, ARINC produced a "Digital Data Compendium", (Reference B) which listed and defined in a uniform way the digital data transmission systems in use at that time. This document brought to the attention of the industry the problems associated with proliferation of data standards. In 1975, the Airline Industry established the Systems Architecture and Interface (SAI) subcommittee of AEEC. One of the tasks of the SAI subcommittee was to establish digital data standards which

would be suitable for the commercial aircraft avionics architecture which was in development. A paper by Hart and Dandakar presented at the 1976 Multiplex Conference (Ref. C), discussed the objectives of the SAI Subcommitte and its work up to that time.

PRESENT STATUS

AEEC has developed an avionic system architecture and prepared "characteristics" for the avionics equipment for new airplane designs. In the area of digital data transmission standards, the SAI subcommittee has developed two specifications covering the needs of the next generation of commercial aircraft. The two buses characterized are; a general purpose bus (ARINC 429), and a very high speed data bus (ARINC 453). The general purpose ARINC 429 bus can operate at one of two specified bit rates, 12 to 14 Kbits/sec (similar to the previous ARINC 575 bus) and at 100 Kbits/sec. The data word labels are allocated and the information bits are defined for each parameter to be sent over the digital data buses for all systems. The bus-transmitter unit terminates the bus, therefore, single bus-transmitter operation only is permitted. ARINC 429 contains any description of the data word necessary to ensure unit interchangeability of the avionics units built to the standard. The description includes such items as maximum time between parameter words, data accuracy, data resolution, and will include parameter transport time and filter parameters necessary to ensure correct operation of the avionics unit in conjunction with other units. Uses of ARINC 429 on new generation aircraft include; digital autopilot, autothrottle, electronic engine controls, navigation, A/N displays, etc. The next generation of commercial aircraft may use approximately 100, ARINC 429 buses per ship set.

During development of the ARINC Weather Radar characteristic (ARINC 708), a requirement arose to transmit digital information from the radar T/R unit to the radar displays, at a bit rate considerably higher than the 100 Kbits/sec available from ARINC 429. The SAI subcommittee initiated development of a "very high speed" digital data bus for weather radar, with the designation number ARINC 453. The subcommittee recommended that, where appropriate, the ARINC 453 bus should use experience and technology from MIL-STD-1553. The weather radar subcommittee has defined a digital data transmission system for the specific task of sending digitized return data to the display unit. The transmission system utilizes MIL-1553 technology and experience in the following areas:

- o Controlled impedance twisted shielded wire.
- o Line terminations and stubbing control.
- o Transformer coupling.
- o Voltage levels, bit rate, rise and fall times, etc.
- o Manchester modulation.
- o Invalid Manchester word sync.

The word structure of MIL-STD-1553 was not adopted due to the specialized weather radar requirements.

At this point in time the requirements for digital data transfer for the SAI defined architecture, can be satisfied by ARINC 429 system plus the ARINC 453 system for weather radar display data.

Unforeseen requirements may evolve for digital data buses during the life of the next generation of commercial aircraft or for future aircraft generations. It is important that the ARINC digital data buses have growth capability to meet these possible future requirements. In the next section of this paper some comment is made on the growth of ARINC 429 and 453 to indicate potential, but not to recommend particular implementations.

## FUTURE GROWTH POTENTIAL: ARINC 429 and ARINC 453

The ARINC 429 general purpose digital data bus system can evolve and grow to accommodate a large spectrum of broadcast applications. The ARINC 429 waveform as presently specified is capable of providing the current necessary to operate opto-isolation LED's. Opto-isolation receivers have the advantage of dielectrically isolating the receiver unit from line voltages. This feature may be useful in a high lightning transient environment. Another growth area now available, subsequent to the allocation of a new group of labels, is multiple word messages.

The ARINC 453 very high speed data transmission system as presently envisioned, is capable for growth in several ways, for example word format and multiple-transmitter bus operation.

The SAI subcommittee has established that for future uses of ARINC 453 (other than weather radar), a 32 bit word format identical to the ARINC 429 word format is to be used, Figure 1. To maintain similarity to the ARINC 429 4 bit interword gap, ARINC 453 has a three bit invalid (high-low) Manchester synchronization pattern followed by a single "dummy bit" (binary "1") to ensure a minimum interword time (sync + dummy) of 4 bits. Future growth could utilize the

remaining 3 sync-plus-dummy sequences to expand the word structure.

## GROWTH OF ARINC 453, MULTIPLE TRANSMITTER OPERATION

The growth of ARINC 453 into multiple-transmitter bus operation is possible on an evolutionary basis. The ARINC 453 electrical characteristics are similar to MIL-STD-1553, therefore, multiple transmitter operation is possible. This feature has been recognized and is discussed in the ARINC specifications.

To permit classification and discussion of multiple transmitter protocols, it is convenient to talk about the "level of complexity" of the bus transmitter unit to support the subject protocol. The "level of complexity" is considered to be in ascending order depending upon the circuit complexity added at the bus-transmitter (bus-TX) unit, over and above the simple bus-TX required for single transmitter operation. The bus-TX unit is considered to include all functions necessary to deliver a message onto the bus.

### BASELINE: Single Transmitter

The bus-TX is free to send a data word when a data word is ready to be sent. The bus-TX has no knowledge of bus status. No data storage is required and no delay (other than normal circuit and clocking delays) is incurred.

### LEVEL 1: No Additions To The Baseline bus-TX

A multiple transmitter protocol can be postulated for this level, where transmitters are allocated exclusive use of the bus by some control element external to the bus. An example is where a system utilizes one of two units as data sources with the power being switched from the main unit to the back-up, in the event of a unit fault. A second example of a protocol at this level is where two-way communication between units has been established on a request/respond basis. In this way each unit has exclusive use of the bus and no clashes result.

Another class of protocol can be postulated for this level, that is to allow two bus-TX's to send simultaneously, assuming that receiver decoders can be constructed to separately decode the simultaneous messages. McEliece and Rubin, (Ref. D), show that under some conditions simultaneous operation of two transmitters can provide a data rate in excess of the time-share protocol.

If the data transmission rate of the two transmitters are plotted as $R_1$ and $R_2$ and the capacity is 1, then the data rates achievable with time sharing the bus lie within the region shaded in Figure 2.

If two unsynchronized bus-TX are permitted to transmit simultaneously on an ARINC 453 channel, then a receiver connected to the bus will sense nine voltage conditions corresponding to the combinations of the three possible transmitter states; HI, LO, OFF.

If it is assumed that two unsynchronized transmitters are continuously transmitting (i.e. transmitting pad words if no data is ready to send), then a receiver will sense only 4 states; a low ("0") when both transmitters are low, a high ("1") when both transmitters are high and an intermediate level ("2") when the transmitter states differ. This condition, (0,2,2,1) corresponds to McEliece and Rubin, channel C. They show that for the C channel, the combined rates are within the area shown in Figure 3, bounded by (0,0; 1,0; 1,1/2; 1/2,1; 0,1; 0,0).

While the use of two transmitter, simultaneous, unsynchronized transmission buses is theoretically possible and capable of higher data rates than time share protocols; its use is difficult due to the wide tolerance permitted on the transmitter output voltages in ARINC 453. It may be possible to redefine the transmitter voltage tolerances and other bus parameters to permit future use of this protocol. This protocol utilizes simple level 1 transmitters but the receivers are complicated by the requirement to decode the resulting composite signals.

A time-shared multiple transmitter protocol can be used at this level if clashes are permitted. Figure 4 shows the probability of clash if n transmitters time-share the bus. The following assumptions are made for Figures 4 to 9, non-synchronized transmitters with the same word length and same word rate.

For the time sharing system here considered, the demodulator is not capable of decoding data during simultaneous transmission by two or more bus-TX, therefore, transmission clashes are considered to produce errors in the demodulators. This protocol could be considered where the probability of clash is so low, that the error control policy can accommodate it. The problem of repeated clash where transmitters are very close to synchronism must be considered.

LEVEL 2: Addition Of Bus Status Detector

The bus status detector is only capable of sensing bus "free" or "busy". The bus "busy" status can be used to inhibit the bus-TX to prevent clashes in a simple two transmitter non-synchronized time shared system. Provision must be made at the bus-TX to store the data word in the event that the bus is busy. This data delay feature is not found in the Level 1 systems or in previous ARINC broadcast systems. The maximum delay suffered, is the word length sent by the alternate bus-TX. The probability of suffering a delay and the average delay is dependent upon the joint bus loading.

If the level 2 protocol is used for three or more bus-TX, the possibility of clashes exist following the condition where two or more bus-TX are delayed awaiting access to the bus. Figure 5 shows the probability of clash and Figure 6 shows the average delay for this protocol.

LEVEL 3: Addition Of Logic

The logic can introduce a pause before a bus-TX sends, following a delay caused by "busy" bus. If this pause is set to a different time for each bus-TX, then clashes will be avoided and the bus-TX with the lowest pause (highest priority) will send its message and cause the other bus-TX's to be further delayed. Figure 7 shows the average delay with this protocol. An alternative is to make the pause time a pseudo-random number; then all bus-TX's are identical and priority is randomly obtained. With the random delay method, a possibility of clash still exists. The probability of clash is shown in Figure 8 and the average delay in Figure 9.

LEVEL 4: Use Of Data Receiver

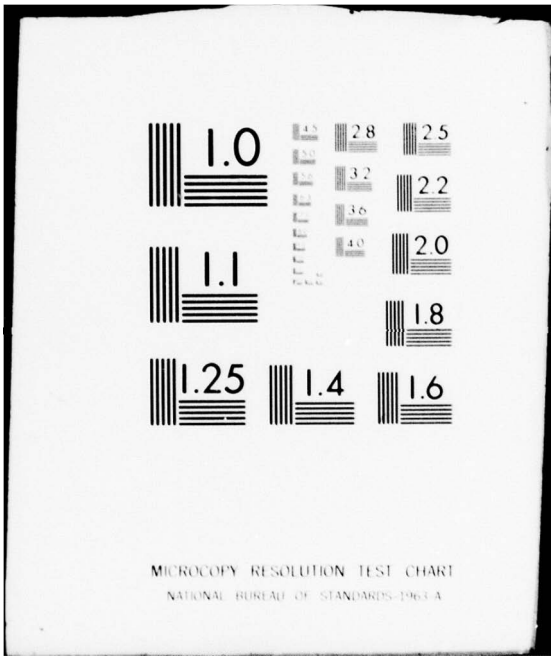
The bus status detector circuit of levels 2 and 3 is replaced with a bus receiver. Logic is then included to extract information of interest from the data being sent by other transmitters. An example of this level would be where each bus-Tx decodes the word labels sent by other bus users. In this way, the bus-TX's could cooperate and send data words on the bus in a predetermined sequence. Appropriate time out and logic could be included to ensure continued operation of the bus, in the event of failure of a cooperating bus-TX. For the protocols discussed to this point all bus-TX units operate autonomously, no bus controller is required.

AD-A068 310 AERONAUTICAL SYSTEMS DIV WRIGHT-PATTERSON AFB OH F/G 17/2
PROCEEDINGS OF THE AFSC MULTIPLEX DATA BUS CONFERENCE (2ND), HE--E
OCT 78 E C GANGL

UNCLASSIFIED ASD-TR-78-34-VOL-2 NL

5 OF 9
AD
A068810




1.0
1.1
1.25
1.4
1.6
2.8
2.5
3.2
2.2
3.6
2.0
4.0
1.8
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

LEVEL 5: Use Of Communication Prior To Sending Message

The highest level considered in this discussion is where a bus receiver and extensive processing capability is included with each bus-TX. The bus-TX's would have the capability to communicate to set up conditions for sending the data message. An example of this protocol is MIL-STD-1553 where one bus-TX is associated with the bus controller function.

An example of an autonomous protocol operating at this level is where the bus-TX's negotiate with each other to determine the message transmission sequence based upon pre-established and/or time-dependent message priorities. The data words would be transmitted in the agreed sequence before further negotiation would take place. In this example, all bus-TX have equal authority on the bus, the sequence of words sent would be a function of message priorities only.

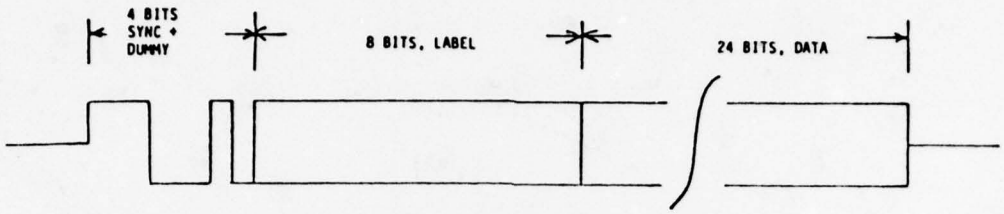
CONCLUSION

In conclusion I would like to mention again the evolutionary aspect of digital data buses on commercial aircraft. The present systems are based upon systems which are in operation. The present characteristics, especially ARINC 453, provide potential for considerable growth to meet possible future requirements. Several interesting aspects of research and development are involved in these growth areas.

REFERENCES


a) Mark 33 Digital Information Transfer System (DITS) ARINC Specification 429-1 June 1, 1975.
b) "Digital Data Compendium" ARINC 419, 1973.
c) Commercial Airline Data Bus Requirements, An Assessment W. G. Harts and A. J. Dandekar, AFSC Multiplex Conference, 1976.
d) "Timesharing without Synchronization", R. J. McEliece and A. L. Rubin, Int Telemetering Conference, 1976.

FIG 1 ARINC 453 WORD FORMAT (OTHER THAN WEATHER RADAR)



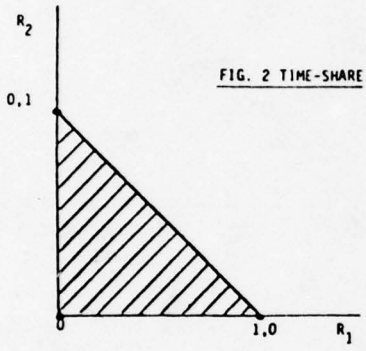




FIG. 2 TIME-SHARE.

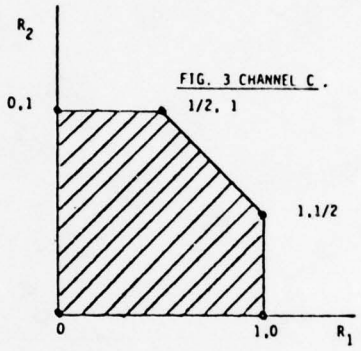



FIG. 3 CHANNEL C.

FIG.4 CLASH PROBABILITY FOR LEVEL 1

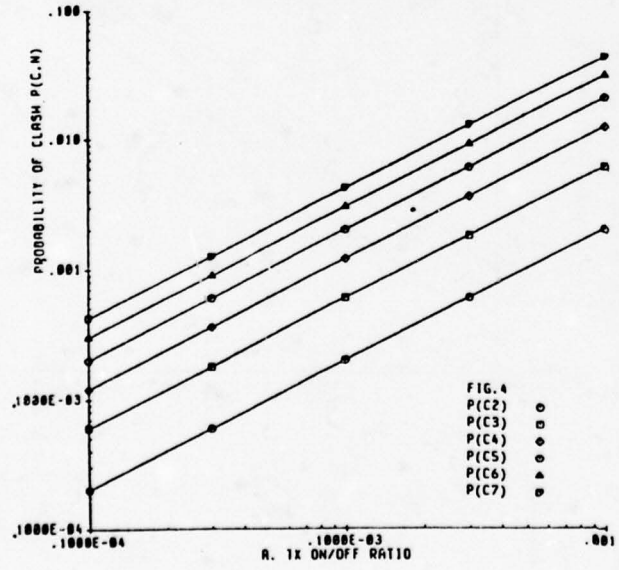



FIG.5 CLASH PROBABILITY FOR LEVEL 2

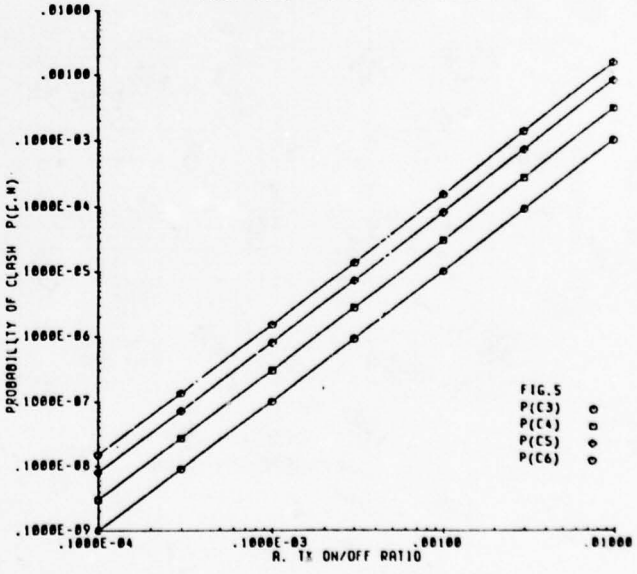


FIG.6 AVERAGE DELAY FOR LEVEL 2

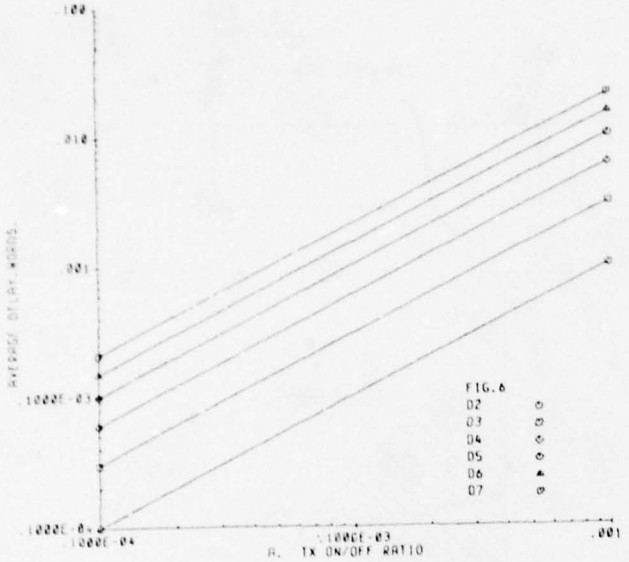



FIG.7 AVERAGE DELAY LEVEL 3
FIXED PRIORITY
4 TX ON BUS

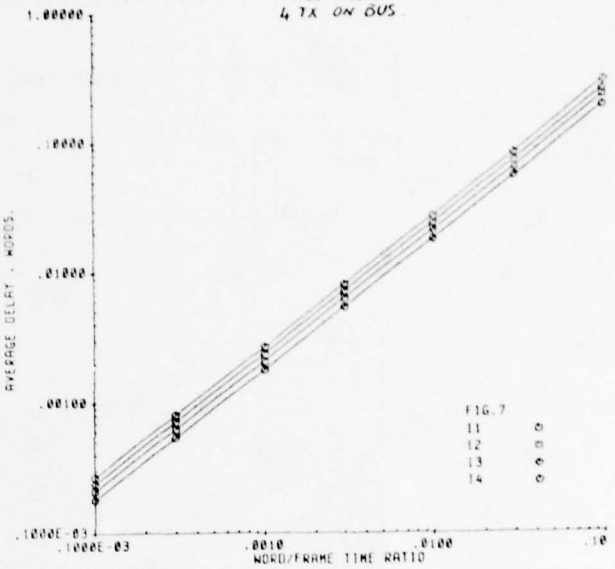
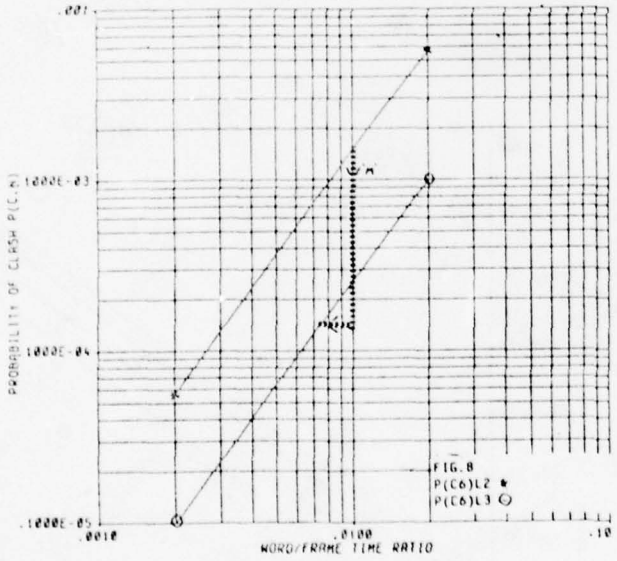


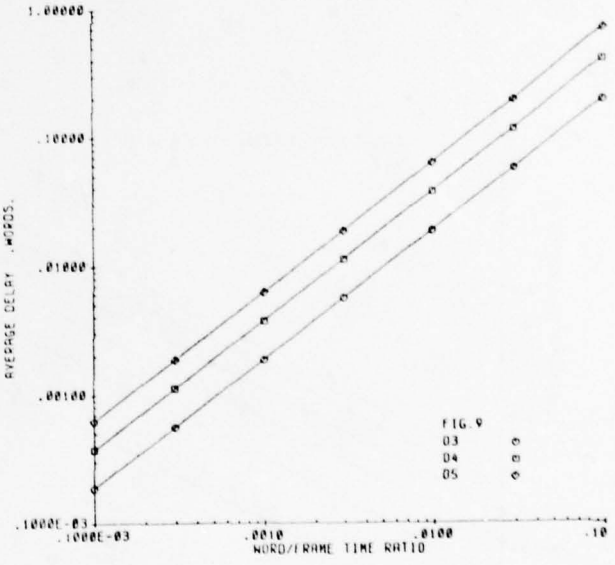

FIG.8 PROBABILITY OF CLASH, LEVEL 3, RANDOM.
APPROXIMATE GRAPHICAL METHOD TO CALCULATE P(C,N)
FROM P(C,N) LEVEL 2




RESOLUTION OF RANDOM PAUSE, M=10

AV PAUSE LENGTH=1/4 WORD

FIG.9 AVERAGE DELAY, LEVEL 3,
RANDOM PRIORITY



# ARINC 575 TO 1553A INTERFACE

John F. Howard

ASD/ENAIB, Wright-Patterson AFB, Ohio

ABSTRACT

This paper deals with the construction of an interface between MIL-STD-1553A and ARINC 575 specified data busses. A comparison of the two bus systems and a general description of the interface operation is presented. Consequences of this particular design and possible areas for improvement are also discussed.

## BACKGROUND

A need for ASD to investigate ARINC 575 came about when the AMST SPO (Advanced Medium STOL Transport System Program Office) saw a potential advantage in using some equipment having a '575 interface. Other equipment would use a MIL-STD-1553A interface for standardization purposes. A question arising from this situation concerned the possibility of interaction between the two busses on the AMST. Also, the need became obvious for ASD engineers to "get smart" about ARINC 575 to allow adequate program management and specification writing.

With this in mind, it was decided to design a simple, quickly built interface between the two busses. A commercial air data computer with a '575 output was located and an operating '1553A bus system was available in the SEAFAC laboratory. Special test equipments (Bus Testers) for debugging '1553A RT's (Remote Terminals) were also available from SEAFAC.

The proposed interface would accept data from either bus and retransmit that data in correct format over the other bus. Data handling would progress only in response to command words sent by the MIL-STD-1553A bus controller. The DEB (Data Exchange Buffer) design would cause '1553A operations to be completely invisible to the ARINC 575 bus. In other words, an instrument with a '575 input connected to this DEB would not see any difference in the data it received, although that data was sent to the DEB over a '1553A data bus.

To accomplish the design objectives, the DEB would be capable of receiving or transmitting on either bus system. Since the two busses operate asynchronously, provision had to be made for simultaneous communications over both busses and for temporary data storage within the DEB. In terms of I/O, the DEB would have two separate ARINC 575 busses (one for receiving and one for transmitting) and two MIL-STD-1553A busses (for a dual redundant system).

## DEMONSTRATION SETUP

To demonstrate the operation of the DEB, the system in Figure 1 was set up. Data received by DEB #1 over a '575 bus is transmitted on command over '1553A. An RT-to-RT transfer from DEB #1 to DEB #2 is used. DEB #2 then retransmits this data over another '575 bus. Even though a double conversion like this is unlikely in an airplane, it shows the ability to convert data received from either bus to the format required for the other bus.

A total of 10 parameters (see Table 1) are output by the air data computer on its '575 bus. Although all of these parameters are retransmitted by DEB #2, an altimeter was the only instrument available with a '575 input. To display all of the parameters, a Hazeltine CRT connected to the '1553A data bus was used. When a transfer from DEB #1 occurs

# DEMONSTRATION SETUP OF
# ARINC 575/MIL-STD 1553A INTERFACE

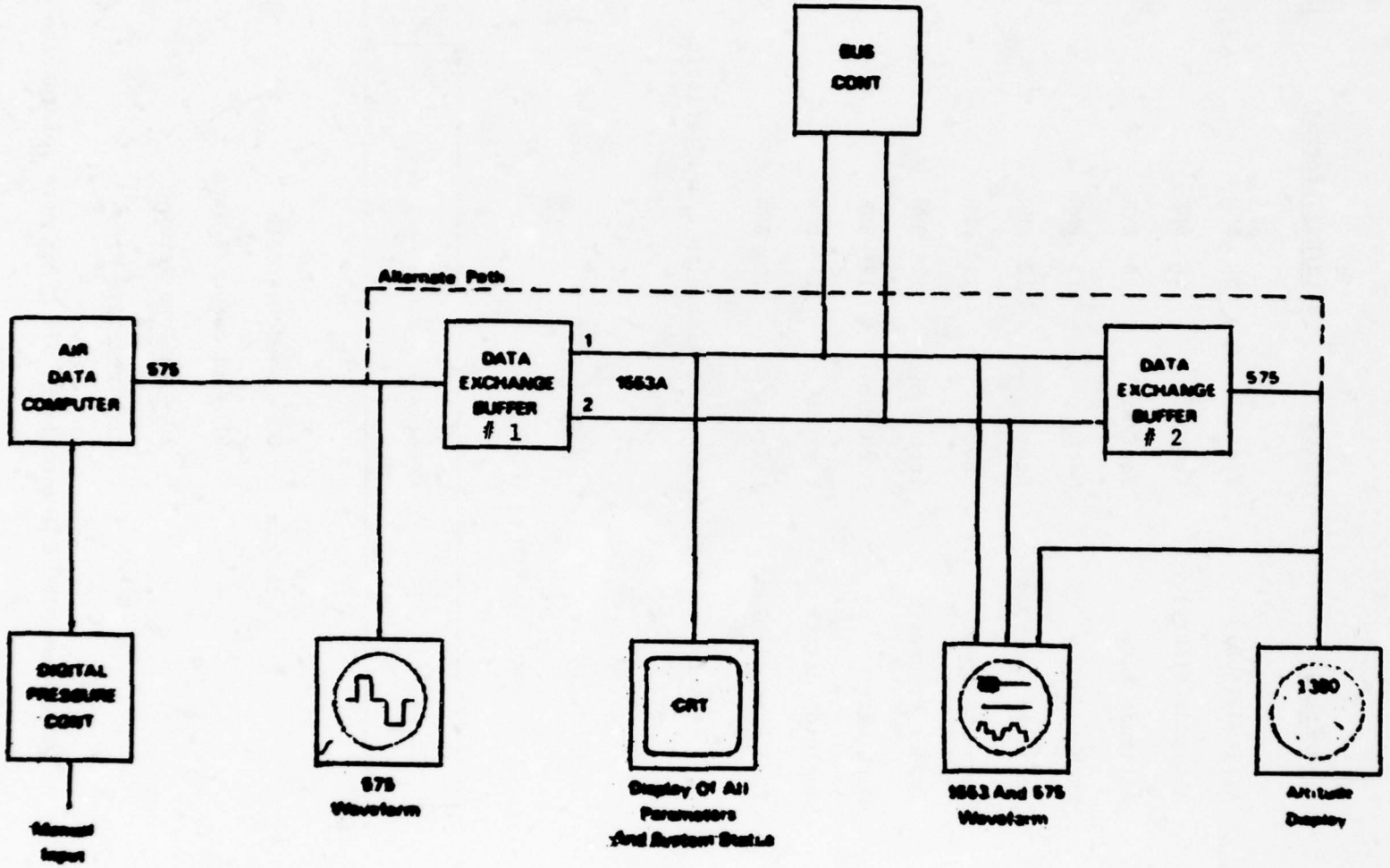


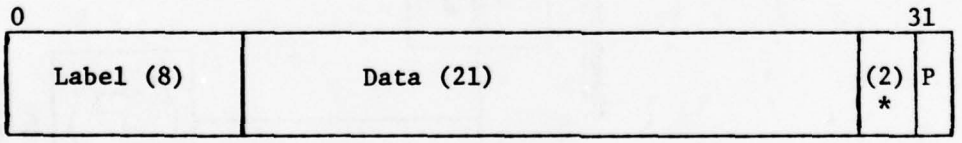

Figure 1

## ADC DATA WORDS

| Parameter | Units | Bits/Dig. Format | Label(octal) |
|---|---|---|---|
| Altitude(BCD) | Feet | 5 BCD | 220 |
| Altitude(Binary) | Feet | 17 BNR | 203 |
| Altitude Rate | Feet/Min | 10 BNR | 212 |
| Mach | Mach | 13 BNR | 205 |
| Computed Air Speed | Knots | 12 BNR | 206 |
| True Air Speed | Knots | 11 BNR | 210 |
| Impact Pressure | Inches Hg | 12* BNR | 216 |
| Mach Rate | Mach/Min | 12* BNR | 214 |
| Angle of Attack | Degrees | 11* BNR | 215 |
| True Free Air Temp | Degrees C | 10 BNR | 213 |

* inferred information, not listed in characteristic

### Word Format




* Matrix 00 - positive data

01 - no computed data

10 - failure warning

11 - negative data

Refer to ARINC Characteristic 575-3 for more information

Table 1

over '1553A, the bus controller also copies the data. Part of the bus controller software identifies the '575 words by their labels and changes the data into ASCII format for the CRT. The bus controller then sends this data to the CRT for display. A typical display is shown in Figure 2.

## COMPARISON OF BUSSES

Although both bus systems communicate via serial data over a single twisted shielded pair, they are different in many respects (see Table 2). Probably the most far-reaching difference comes under the heading of "Protocol." The '575 bus operates in a broadcast mode, i.e., a single transmitter sends data to a maximum of 20 receivers. Only one transmitter is allowed on a bus; therefore, multiple busses are needed to allow for multiple transmitters. Any unit receiving data from more than one source is required to have a separate input for each source. Since communication is one-way only, no feedback concerning the status of the receivers is allowed. Also, any garbled messages will be lost since the receivers cannot request a repeat. An advantage of the '575 bus over '1553A is that the receivers and transmitters are significantly simpler in design and much slower speeds are required. Other differences between the two systems are summarized in Table 2.

## DEB OPERATION

The DEB unit incorporates two logic cards: an MTU (Multiplex Terminal Unit) and an SSIU (Sub-System Interface Unit). The SEAFAC laboratory had available an already proven general purpose MTU design. This MTU communicates directly over a dual redundant '1553A bus and provides a subsystem with buffered data and various control signals. To shorten and simplify the DEB design, this MTU was used and only the SSIU had to be designed.

The SSIU design is divided into a '575 receiving section, a '575 transmitting section, and an MTU interface. Control lines and a clock from the MTU are used to control the SSIU. Incoming 32-bit data words from the ARINC 575 bus are accepted by the "receiver" and stored as two 16-bit words in preparation for '1553A. On command from the MTU (derived from a '1553A command word), these stored data are transferred to the MTU and then out over the MIL-STD-1553A data bus. When data are to be received over '1553A for transmission over '575, the MTU commands the '575 "transmitter" (again derived from the '1553A command word) to accept data. Groups of two 16-bit words are concatenated by the DEB into a single 32-bit ARINC 575 word and transmitted over the outgoing '575 bus.

The software resident in the '1553A bus controller performs a number of jobs. At start-up, a communications check is made of the two DEB's and the '1553A bus. When the communications link has been validated, the software commands RT-to-RT transfers between the two DEB's, takes a copy of the data transferred, puts the data into ASCII format and transmits the data to the Hazeltine CRT display. Each RT-to-RT transfer consists

## DATA BUS DIFFERENCES

| | MIL-STD-1553A | ARINC 575 |
|---|---|---|
| PROTOCOL | COMMAND/RESPONSE | BROADCAST |
| ACCESS | MULTIPLE | SINGLE |
| DATA IDENTIFICATION | TERMINAL ADDRESS | WORD LABEL |
| DATA FORMAT | USER DEFINED | PREDEFINED |
| ERROR CHECKING | FILTER | FILTER |
| | BIPHASE | PARITY |
| | PARITY | |
| | WORD COUNT | |
| | USER DEFINED | |
| ERROR CORRECTION | RETRANSMISSION | NONE |

Table 2

of a 20-word message (10 ARINC 575 words) and occurs every 58 ms. Data are sent to the CRT and refreshed every 464 ms.

## CONSEQUENCES OF DESIGN

The DEB was designed to have no intelligence. Operation is in reaction to activity over the '575 bus or to commands sent over the '1553A bus. This approach made it possible to build and debug the DEB within a very short amount of time. There are, however, some operations an intelligent interface could perform that the present DEB does not perform. In the present setup, the bus controller software must identify each '575 word before reformatting for the CRT display. An intelligent interface could recognize the labels of incoming ARINC 575 data. This information could either be transmitted to the bus controller or the data could be arranged in a preset order, thus relieving the bus controller software of one task.

Parity checking is used on all '575 data except BCD encoded words. Rather than have no error detection at all, the DEB checks for odd parity and rejects all words with even parity. Because of this, the DEB throws out many "BCD Altitude" words in the demonstration setup (see Figure 2). Other interfaces could be made to recognize the labels of BCD encoded words and not reject them for even parity or perhaps not check for parity at all.

The DEB as designed is highly dependent on the '1553A update rate. When the bus controller does not take data fast enough, new data incoming on the ARINC 575 bus is ignored and lost. If the bus controller causes data to be transferred to a DEB too fast, some messages may be lost since the buffers will fill up. In either case, this interface design can cause new data to be discarded in favor of old data. A number of solutions for this problem exist including extra buffering, intelligence, etc., but were not considered worth the time and effort required in light of the objectives of this project. The "easiest" solution was used, namely not allowing the problem to occur. The data rate from the air data computer was measured and used to determine the rate of transfer over '1553A. This solved the potential problem in our simple demonstration setup, but may be inadequate for more complex systems.

## CONCLUSIONS/LESSONS LEARNED

The DEB was designed in the easiest and quickest manner possible. This agreed with the project objectives which did not necessarily include an elegant or flightworthy design. Since a general purpose MTU was used, many inefficiencies exist in the areas of buffering, data massaging, and handshaking within the DEB. A design from scratch, although more efficient, would have taken too long. Note one important fact: the DEB was designed as a separate unit only for this demonstration. In an aircraft, these interfaces would normally not require a separate LRU.

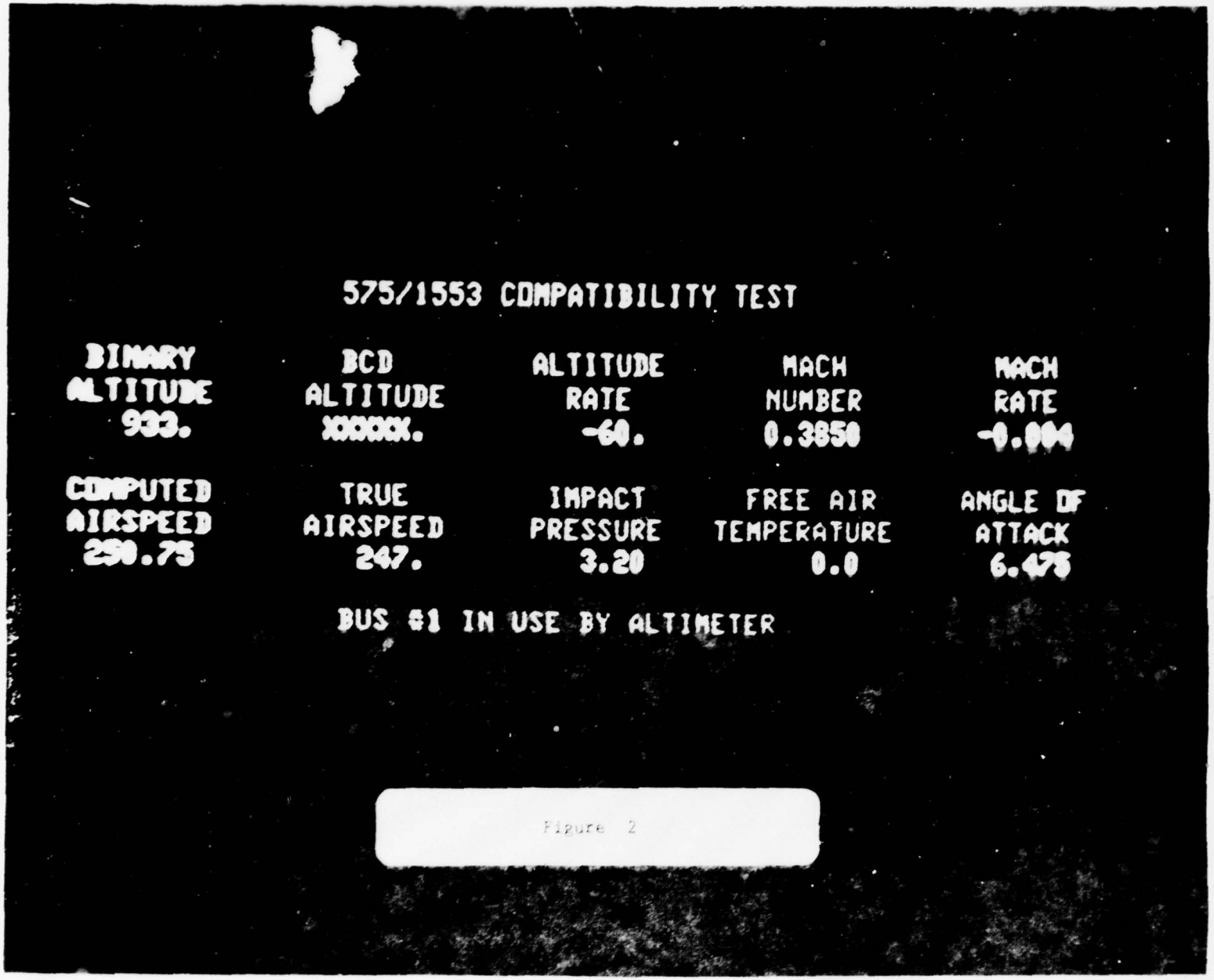

575/1553 COMPATIBILITY TEST
BINARY ALTITUDE 933.
BCD ALTITUDE XXXXX.
ALTITUDE RATE -60.
MACH NUMBER 0.3850
MACH RATE -0.004
COMPUTED AIRSPEED 250.75
TRUE AIRSPEED 247.
IMPACT PRESSURE 3.20
FREE AIR TEMPERATURE 0.0
ANGLE OF ATTACK 6.475
BUS #1 IN USE BY ALTIMETER

Figure 2

Each bus system carries along a certain amount of overhead in addition to the transmitted data. This overhead includes such items as labels, command and status words, interword or intermessage gaps, parity bits, etc. In the demonstration system the entire ARINC 575 word as received is transmitted over '1553A. Overheads including the label, sign status matrix, and parity originating in the '575 bus are carried over and added to the load carried by '1553A. The MIL-STD-1553A bus handled the load easily in this demonstration, but a more heavily loaded bus might run into difficulty. Stripping unnecessary bits from a word and packing words together could be done if a more efficient transmission were required. If this were done, however, extra intelligence/software would be required at the receiving end.

This interface design achieved its purpose. The DEB successfully allows communication between an ARINC 575 bus and a MIL-STD-1553A bus. Design inefficiencies exist, but no serious interfacing problems were encountered. Through the process of building the DEB, ENA engineers learned about ARINC 575 and found that only a relatively simple design was required to interface the two bus systems.

REFERENCES

1. MIL-STD-1553A, Aircraft Internal Time Division Command/Response Multiplex Data Bus, 30 April 1975

2. ARINC Characteristic 575-3, Mark 3 Sub-Sonic Air Data System (Digital) DADS, 15 July 1971

3. Dual Channel Multiplex Terminal Unit Hardware Documentation Package, August 1977

# BUS CONTROLLER SOFTWARE IMPLEMENTATION

Thomas L. Zawodny
ASD/ENAIA/(SEAFAC)
Wright Patterson AFB, OH 45433

## ABSTRACT

This paper deals with the complexity of implementing the bus control function into a system. After defining the bus control function, various bus controller designs are examined with emphasis on the software requirements of each. General performance type considerations are discussed leading to the conclusion that definition of the bus control function is a system's level problem.

PRECEDING PAGE BLANK

## THE BUS CONTROL FUNCTION

Continuing advances in the state-of-the-art in digital electronics are causing significant impact on the numbers of avionics equipment being proposed for, and implemented into, future and existing weapons systems. Entire subsystems such as stores management, fire control, and navigation are being designed with embedded minicomputers, and/or microprocessors in addition to digital electronic components. Digital electronics is coming of age in avionics!

New systems, and old system updates, reveal an increasing data interdependence between subsystems, sensors, and/or "functions". Interchange and transfer of much of these data between "units" is being accomplished via the multiplex data bus. Controlling the flow of these data over the data bus is the function of the bus controller.

## HARDWARE/SOFTWARE CONTINUUM

The design and implementation of a bus controller generally embodies a blending of hardware and software; hardware for its inherent reliability and software for its attendant flexibility. During the initial design phase conflicts may ensue between hardware and software designers when a simplification by one causes complication for the other. Typically, though, the hardware design is completed first and any resulting complications or deficiencies are left for software to resolve. The degree of software needed to implement the bus control function (message sequence and timing) will depend on the basic hardware approach to the system architecture.

Let us examine three candidate families of bus controller designs and highlight the software needed to implement each.

A. Programmed Hardware - See Figure 1. In this bus controller design, the function to be performed is pre-determined and programmed onto ROMs or PROMs. The circuitry within the bus controller accesses these (P)ROMs and initiates the data flow across the bus to the required units (remote terminals). When the function to be performed changes (i.e. when the system requirements change), (P)ROMs with the new schedule of data transfers are programmed to replace those existing within the bus controller. The software, then, is mainly limited to that necessary for programming of the (P)ROMs.

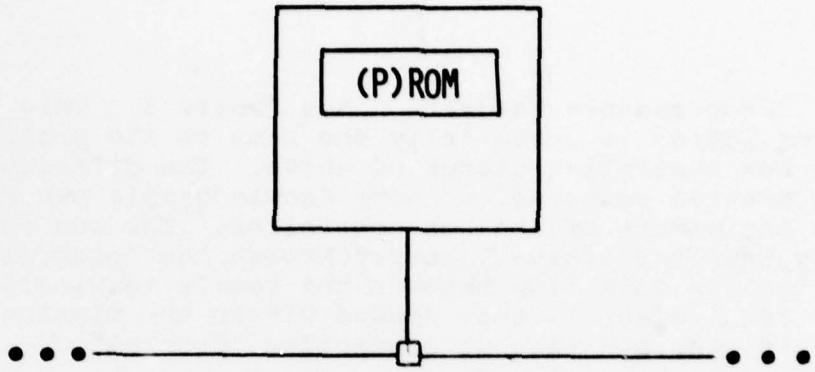



FIGURE 1. PROGRAMMED HARDWARE BUS CONTROLLER

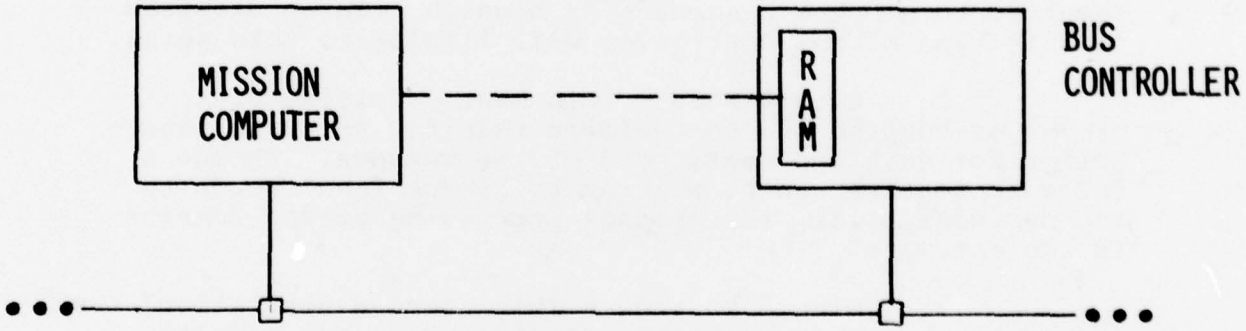



FIGURE 2. PROGRAMMABLE HARDWARE BUS CONTROLLER

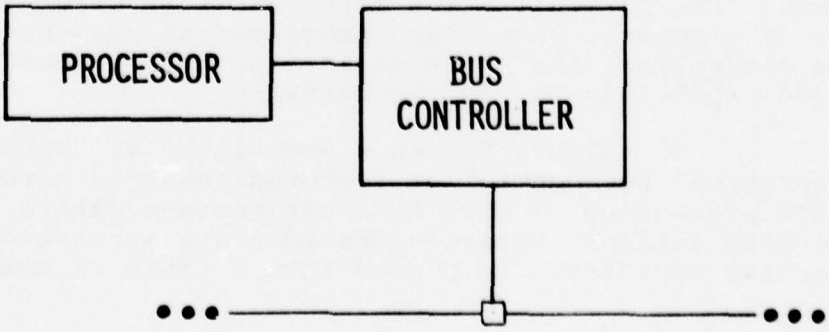



FIGURE 3. PROCESSOR-COUPLED BUS CONTROLLER

B. Programmable Hardware - See Figure 2. This bus controller design is essentially the same as the programmed hardware bus controller discussed above. The difference is that the mission computer, or some knowledgeable processor, programs the memory of the bus controller. The bus controller circuitry then repetitively steps through the "program" thus controlling the data flow between the remote terminals. The software requirement is that needed within the mission computer to generate the bus controller "program".

C. Processor-Coupled Bus Controller - See Figure 3. In this configuration, the bus controller is linked to a processor which executes a software program to oversee the bus control function. This processor may be dedicated to the bus control function or it may have additional processing requirements. The software complexity to implement this bus controller design is totally dependent on the sophistication of the bus controller circuitry itself. That is, the more primitive the circuitry, the greater the software interaction required to process a command or message. Three examples of this type of bus controller will illustrate this point.

1. Single Word - This most primitive of processor-coupled bus controllers requires software interaction for each and every word of the message. Though a software routine can be written to "force feed" words to the bus controller, the message processing burden remains in the software.

2. Single Message - This "second generation" bus controller has the capability of processing one complete message at a time. The processor software sends the starting memory address of the message to the bus controller which, in turn, performs a direct memory access (DMA) into the processor memory for the required message. The bus transaction is then processed under direct control of the hardware, which signals the software when the activity finishes. The software is then left to examine the returned transaction status information (status word, etc.) in order to insure that the message was indeed processed as requested. In this bus controller, the software interaction is lessened by the added capability within the hardware.

3. Table Driven - See Figure 4. This "third generation" bus controller features enhanced hardware to allow processing of more than one message with a single software action. Numerous messages are structured and their starting addresses are placed into a table in memory. To

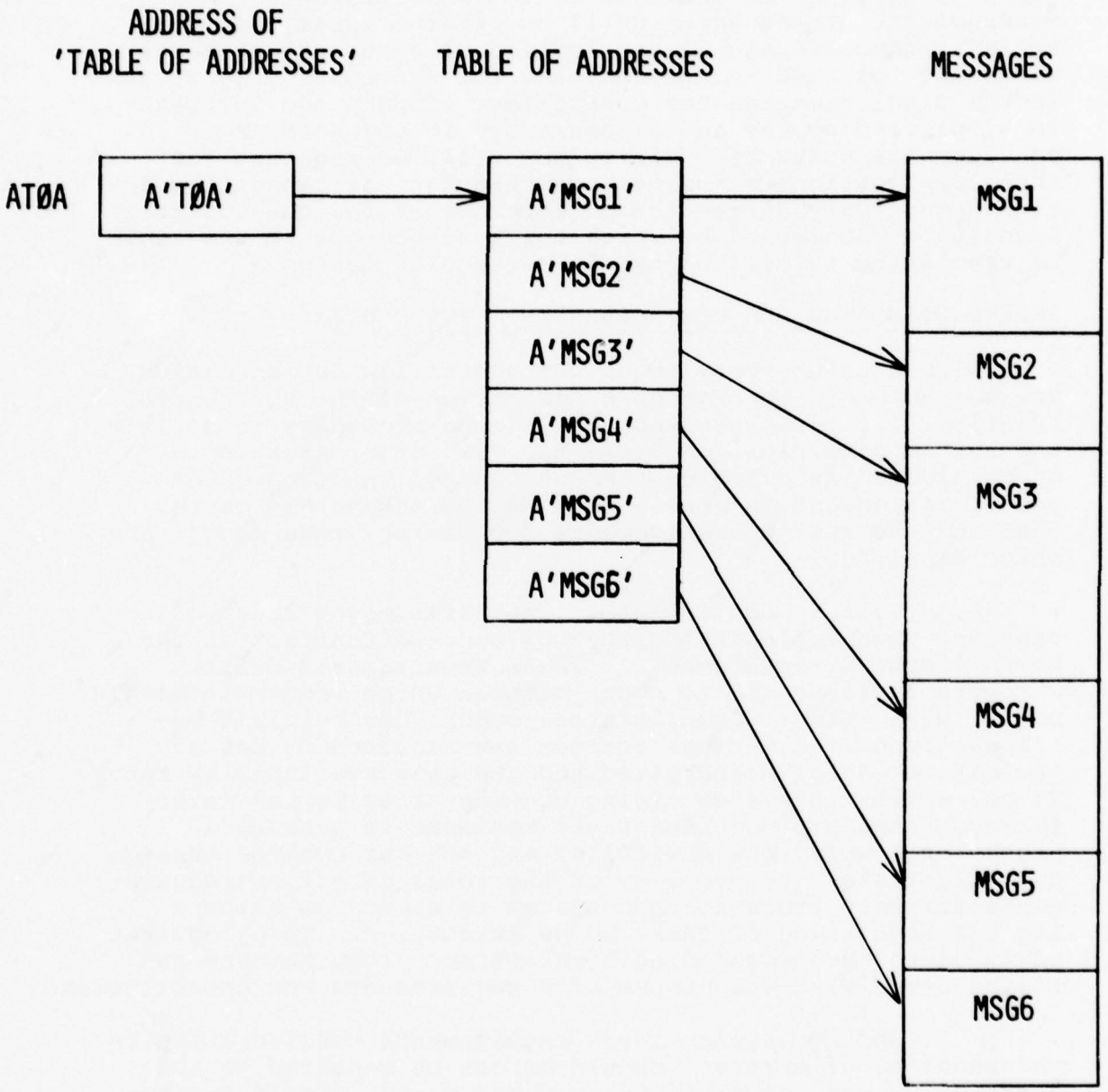



FIGURE 4. TABLE-DRIVEN, PROCESSOR-COUPLED BUS CONTROLLER MESSAGE TABLE STRUCTURE

initiate processing of these messages, the starting address of the table of addresses is passed to the bus controller. The bus controller continues DMAs into the processor memory, stepping through the table of addresses until either all messages are processed or until an error occurs. An activity-complete signal and individual transaction status words are returned to the software for examination, just as in the single message bus controller. Though the software is simplified by the added complexity in the hardware, considerable software activity may still be required for those applications where message structure or table organization must vary during the performance of the bus control function. The method by which the last message in the table is recognized is defined by the particular design.

## IMPLEMENTING THE BUS CONTROLLER INTO THE SYSTEM

With a brief overview of bus controller design behind us, we can begin to examine a definition of the bus control function; i.e., the sequence and timing necessary to satisfy the system data flow requirements. Two major areas need to be considered in defining the bus control function: the peculiarities and interrelations of the subsystems on the bus, and the system requirements for timing, redundancy, and error handling.

A. System Requirements. The first major area to consider when implementing the bus control function is the overall system requirements. These requirements define what transactions are to occur between which remote terminals and at what rates. Complications occur when multiple bus transactions from various sources are required to satisfy the data needs of a subsystem and the data availability rates do not match. A system timing consideration is the interference between processing tasks resident in a shared processor-coupled bus controller and the bus control message queueing tasks. An overview of the total data flow requirements for both processor and system is needed to allow a logical sequencing of tasks to be structured. This requires an in-depth knowledge of each subsystem's requirements and timing along with the processor's requirements and capabilities.

Another system-level requirements consideration is the handling of errors. Should errors be reported to the operator or executive; if so, how and when? Should the bus controller be smart enough to retry an offending message; if so, how many retries? Bus transaction errors (no response,

bad-sync, etc.) are easily detected by hardware. However, an error also occurs when the data "received" does not match the data "sent". Should this error be anticipated and corrected, and if so, how?

A last systems-level consideration is redundancy, of both buses and bus controllers. Should traffic occur on both buses "simultaneously" or should only one be active? In a redundant bus controller system, how is failure of the primary controller detected, who detects it, and how is the back-up activated?

B. Subsystem Peculiarities. Though it is desireable for the various remote terminals on the bus to possess identical capabilities, a particular system may not have that luxury, particularly in system updates. For example, does the readiness of a remote terminal need to be verified before initiating a bus transaction? Some designs provide this feature while others do not. An exchange of data between two remote terminals will be complicated if one or both have only receive/transmit capabilities as opposed to terminal-to-terminal transfer.

## CONCLUSIONS

The implementation of a bus controller function is a highly complex problem requiring resolution at the systems design level.

The system design process generally begins by defining the purpose of the system and by specifying the system performance requirements. When the system requirements have been defined, various subsystems are proposed to meet them. Further detailed study refines the definition of the subsystems, including the interdependent data flow and timing requirements of each. The combination of these requirements becomes the definition of the bus control function. The realization of this function then becomes the design effort of hardware and software combined.

It is important to realize the pivotal role the bus controller plays within a system. The capabilities of various subsystems are multiplexed for increased system flexibility and more centralized control. The bus control function can be viewed as the subsystem integrator or as the "glue" that holds the system together.

# AN AIRCRAFT COMMUNICATIONS MANAGEMENT TERMINAL (CMT)

by

T. Gedeon and S. Thornton, SCI
Systems, Inc., Huntsville, AL

(This paper was not submitted)

PRECEDING PAGE BLANK

MULTIPLEX DATA BUS CONFERENCE

10-12 OCTOBER 1978

SESSION IV

NEW DATA BUS CONCEPTS - PART II

MODERATOR: Dr. William E. Eppers, Jr.
Deputy Director
AFAL, W-PAFB, OH



PRECEDING PAGE BLANK

# MULTIPLEX CONSIDERATIONS AND POTENTIAL APPLICATIONS IN MODULAR AUTOMATIC TEST EQUIPMENT (MATE)

John E. Scott

SPERRY SYSTEMS MANAGEMENT
Sperry Rand Corporation
Great Neck, N.Y.

ABSTRACT

This paper describes the MATE System program and discusses the multiplexing approaches that will be examined. The MATE System program output will consist of four "guides" containing standards, specifications, manuals, handbooks, and procedures. Included in these guides will be a set of interface standards that makes maximum use of readily available test equipment modules and standard data buses. The elements to be considered in developing these interface standards are described, and the principal characteristics of several candidate bus standards are discussed.

## A. THE MATE PROGRAM

### 1. General

The Modular Automatic Test Equipment (MATE) program has been conceived by the Air Force as a means of providing effective support of future avionics equipment at the lowest possible life-cycle cost. This program has been divided into several segments, the first of which, the MATE System program, was started in July of this year, and will be completed in December 1980. The next segment of the program will improve and refine the concepts developed during the first segment, and will implement the resulting improved concepts in the form of a prototype MATE system.



PRECEDING PAGE BLANK

The specific objectives of the overall program are defined as follows:

- Reduce life-cycle costs of weapon system support, including costs of automatic test equipment (ATE).
- Reduce the number of different types of ATE required.
- Improve operational utility, test efficiency, and management of ATE.
- Improve ATE procurement practices.

These objectives are guiding the activities of the current MATE System program.

2. MATE System Program Outputs

The end products of the MATE System program will consist of four guides, each of which will include standards, manuals, handbooks, and management procedures. In each guide there will be references to the other guides, to data systems, and to other standards that form an integral part of the design comcept. The four guides are titled: Electronic Test Equipment Acquisition Guide, MATE Development Guide, Avionics Testability Design Guide, and Production/Operational Guide.

The Electronic Test Equipment Acquisition Guide will provide information for acquiring effective low-cost support systems. This information will define responsibilities and decision points for equipment acquisition, and will include procedures for making life-cycle cost trade studies and for matching test system changes to weapon system changes.

The MATE Development Guide will provide information for the design of MATE test configurations to meet specific performance requirements. This guide will cover all aspects of the system, including hardware, software, and human interfaces, and will be based on requirements of avionics units to be tested in accordance with the recommended maintenance plan.

The Avionics Testability Design Guide will provide guidance and checkpoints to the avionics design engineers for incorporating design features that facilitate testing the units.

The MATE Production/Operational Guide will contain information required for using MATE test systems to maintain configuration control and to establish feedback for keeping MATE systems responsive to operational needs.

These guides will provide the technical tools and procedures for acquisition management of avionics support, and for development, design, production, and logistics support of future automatic test systems.

3. Program Tasks

The system development program is divided into three principal phases: the Survey/Study Phase, Study/Verification Phase, and Demonstration Phase.

The Survey/Study Phase includes the following tasks:

- Perform surveys of operational sites, test equipment manufacturers' facilities, and literature pertaining to all aspects of ATE.
- Reduce data obtained from the surveys.
- Perform initial studies and analyses of hardware, software, human interfaces, testability, and avionics support system acquisition.
- Conduct design studies to synthesize and analyze candidate system designs.
- Formulate optimal support equipment acquisition approaches and procedures.
- Produce draft copies of four program end product guides.

The Study/Verification Phase includes the following tasks:

- Verify the content of the four draft guides, where necessary, through procurement and phototyping of hardware, software, or documentation items needed to verify advanced design concepts.
- Prepare final draft copies of the four guides.
- Develop a test and evaluation plan for the Demonstration Phase.

The Demonstration Phase includes the following tasks, which will be accomplished through analysis, simulation, and test.

- Demonstrate how use of the four guides would have enhanced testability of selected avionics units.
- Use the guides to define a test system for support of 15 avionics units selected by the Air Force.
- Use the guides to design, fabricate, operate, and evaluate four test station configurations based on requirements of four avionics units selected by the Air Force; and demonstrate fault isolation and modularity of the test stations.
- Update the four guides, as necessary, based on the results of the demonstration.

## B. INSTRUMENTATION CONCEPTS

The study of various candidate support system approaches during the first two phases of the MATE System program will include developing an optimum set of interface standards that make maximum use of available military and commercial test equipment. Such interface standards will permit configuring of modules from various manufacturers into automatic test systems. Also, these standards

will permit substitution of modules from different manufacturers for testing identical types of avionics units. They will also permit reconfiguring the system for testing different types of avionics units. The standards will be adaptable to new avionics and test equipment designs, and will be compatible with industry standards wherever possible.

The relationships among the elements to be considered in developing these interface standards are illustrated in Figure 1, which shows a simplified block diagram of a typical test station. This station contains a controller (which may be a digital computer and peripherals), interface adapters, multiple test modules (stimulus and measurement test instruments), the unit under test, and the interconnecting buses. Multiplexing of data is required to provide control and response signals between the controller and the various test modules, as well as between the test modules themselves.

The development of these interface standards will include examination of the following items.

- Requirements of avionics units to be tested
- Human interface requirements
- Controllers and test equipment modules capable of meeting these requirements
- Different types of multiplex data buses, including both standardized and unstandarized types
- Interface adapter designs needed to match the controller and test modules to the bus characteristics

These items are discussed in the following paragraphs.

The requirements of avionics units to be tested will be obtained from the results of the surveys and studies conducted during the first phase of the program. All

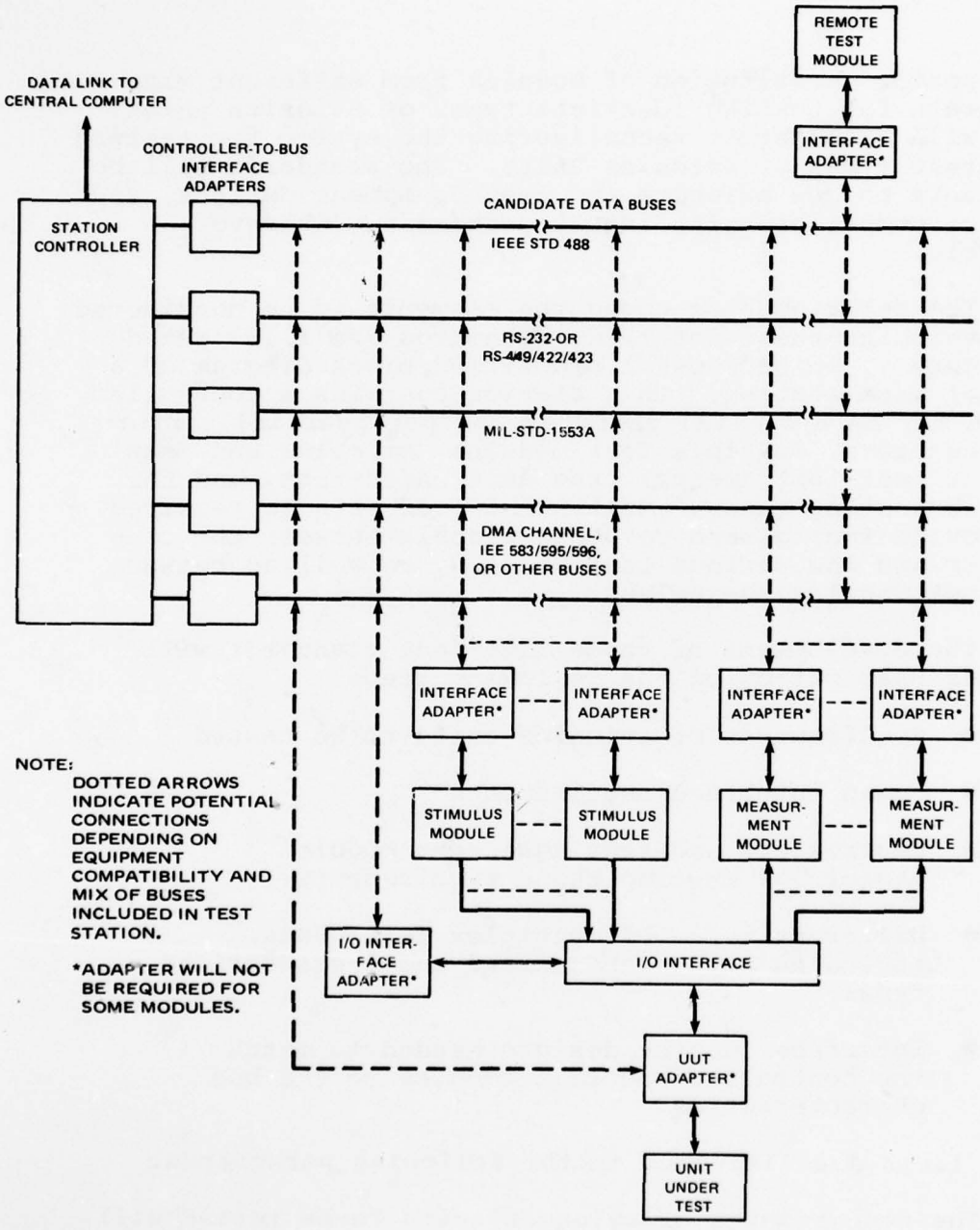



Figure 1. Typical Test Station Interconnections

classes of existing and proposed avionics units will be included (e.g., digital, analog, r-f, microwave, and optical, with many types in each class).

The human interface requirements will also be obtained from the surveys and studies. They will influence the choice of candidate hardware, software, and data bus elements by defining display, control, and equipment-handling features.

Controllers and test equipment modules available for meeting these requirements may differ in their interface characteristics. Typical differences among these elements can be summarized as follows.

- The data type may be completely serial or combined serial and parallel.
- Data rates may vary from zero to several megabits per second.
- Data formats may include various methods of synchronization, addressing, data organization, coding, and error checking.
- The number and timing relationships of control signals for handshaking and interface management may vary.
- Signal characteristics may differ in amplitude, waveshape, or polarity.
- Source or load characteristics may vary in impedance, allowable voltage limits, or sensitivity to noise.
- Mechanical features may vary, such as connector types, connector locations, or cabinet dimensions.

The interface adapters, interconnecting buses, and controller software must accommodate these differences between the various candidate controllers and test equipment modules, and must provide optimum test capability. Thus, in configuring various hypothetical ATE systems as

part of the MATE interface standards development process, trade-offs among the following candidate choices must be considered:

- Number of different types of avionics units accommodated
- Number of different controllers and test modules
- Complexity and number of interface adapters
- Number of different types of buses and number of buses of the same type
- Degree of data processing at the test module to reduce bus data transmission rates
- Degree of modification of standard test equipment modules to reduce number or complexity of adapters
- Level of test equipment testability and self-test

The goal of this trade study is to define a set of interface standards that provide optimum mixes of adapters, buses, software, controllers, and test equipment modules for all foreseeable test situations.

## C. CANDIDATE DATA BUSES

### 1. General

In order to accommodate the differences in data rates, and to permit a wide choice of controllers, test modules, and interface adapters, a number of different bus types will be considered for inclusion in the MATE interface standards. The significant characteristics of several candidate buses are discussed in the following paragraphs.

### 2. IEEE Standard 488-1975

The IEEE Standard Digital Interface for Programmable Instrumentation (IEEE Std. 488-1975*) has gained wide

- - - - - - -

*Also known as ANSI MC 1.1-1975, or HP-IB, or GPIB.

acceptance in the test equipment industry. The intended use of this bus standard is to define the interconnection of both programmable and nonprogrammable devices assembled as an instrumentation system.

The principal features of this bus include: capability of handling up to 15 devices on a single bus; transmission of data as a sequence of 8-bit bytes on 8 wires (bit-parallel/byte-serial); transmission of control signals on 8 wires; and operation at a maximum data rate of one megabyte per second over short distances.

This bus standard is expected to be compatible with most of the applicable types of test equipment modules and controllers that will be available for implementation on MATE.

3. EIA Standards RS-232-C and RS-449

EIA Standard RS-232 has been the accepted standard for interfacing serial data communication equipment and computers with data terminals (such as teletypes and CRTs) for many years. The newer EIA Standard RS-449 together with EIA Standards RS-422 and RS-423 are intended to gradually replace RS-232-C (the latest revision of RS-232).

The RS-232-C standard specifies transmission of data serially between two devices at rates up to about 20,000 bits per second. Fourteen control signals are defined, but fewer are required in most applications.

The combined RS-449, RS-422, and RS-423 standards have been issued to specify new electrical characteristics and to define several new interface functions. The new electrical characteristics accommodate advances in integrated circuit design, reduce crosstalk, permit greater separation of equipment, and permit higher data signaling rates (up to 2 megabits per second maximum).

4. MIL-STD-1553A

The MIL-STD-1553A multiplex data bus standard defines the requirements for data communications among various avionics units within an aircraft. Because both existing and future avionics systems will be compatible with this bus, it is expected that this interface will be useful for test purposes. For example, the test equipment controller could feed data representing simulated sensor values to the

digital computer in a unit under test and receive resulting outputs from the unit. The bus might also be used with test equipment modules designed to operate with this interface.

5. IEEE Standards 583/595/596 (CAMAC)

The CAMAC interface standard has been used in nuclear laboratories and process control industries for several years. This standard defines in detail the mechanical construction of a rack-mounted chassis called a "crate" which houses a controller and up to 23 test modules or adapters. It also defines methods of interconnecting multiple crates.

The bus provides 83 wires within the crate, including 24 data wires in each direction. Data may be transmitted between crates in either serial or parallel form at rates up to 5 megahertz.

6. Other Interface Candidates

Types of interfaces other than those summarized in the preceding paragraphs may be included in the interface standards being developed on the MATE program. For example, if data rates are too high to be conveniently accommodated by these buses, or by multiple sets of these buses, a controller that includes a computer having a direct memory access (DMA) channel may be required.

The decisions regarding which buses to include in the MATE interface standards will result from consideration of the various existing and future data requirements, the characteristics of available equipment, and the feasibility of distributing processing functions.

D. CONCLUSIONS

The multiplexing aspects of the MATE System program require the integration of test equipment hardware and software elements with the characteristics of various data buses to provide a set of useful interface standards. These standards must be flexible enough to meet requirements of future test equipment and avionics equipment, and yet provide sufficient detail for assuring successful interchangeability of modules. The Air Force has directed that all aspects of this study be examined from a systems viewpoint, thus maximizing the probability that these requirements will be met.

# DISTRIBUTED PROCESSING FOR MIL-STD-1553A

Carl Wise
Westinghouse, Inc.
Baltimore, Maryland

James Bain
Air Force Avionics Laboratory
Dayton, Ohio

## ABSTRACT

This paper advocates a more sophisticated distributed processing approach than is currently used in MIL-STD-1553A. Rather than a central bus controller, distributed control with "smart" terminals can provide a higher degree of reliability and adaptiveness to avionics systems. A new provision in the proposed MIL-STD-1553B opens this possibility by defining a dynamic bus allocation protocol. A candidate approach for distributed control is presented.

## DISTRIBUTED PROCESSING TRENDS IN AVIONICS

A recent study indicates that distributed processing will be the most dynamic segment of the computer industry in 1981[1]. The underlying force behind this change is, of course, the microprocessor revolution. Low cost distributed intelligence will have a strong influence on system design for many years to come. This influence is evident in recently developed avionics systems. The F-15 and F-16 have several processors dedicated to separate functions such as navigation, radar, and fire control, and are interconnected by a 1 MHz time division multiplex data bus. This is in contrast to aircraft designs of the 1960's which generally have one large computer and no multiplex data bus. The distributed processing trend is expected to continue in future aircraft with large numbers of microcomputers performing dedicated functions in an interconnected network. These processors will operate nearly autonomously and communicate via a shared resource - the avionics data bus.

A key issue in the design of future avionic distributed processing systems will be how to control allocation of the data bus and processing resources. In systems with one computer, the central control concept obviously applies. This concept can even be extended to systems with several computers by partitioning software tasks among processors in advance, and by assigning one as the master bus controller. This technique forms the basis of MIL-STD-1553 and 1553A[2] in which the bus control computer is programmed in advance with all synchronous bus transmissions. The central control concept works well in an avionics system with several minicomputers and "dumb" remote terminals. However, the authors feel it is a suboptimal approach in future designs which will be characterized by smaller, more numerous processors and "smart" remote terminals. New approaches have been developed in which responsibility for data bus and resource allocation is shared by multiple computers. These approaches have been collectively termed "distributed control," and have potential to provide many advantages, most notably adaptiveness.

## DISTRIBUTED CONTROL CAN BE ADAPTIVE

The emphasis in military avionics in the future will not be in simply interfacing equipment, but in achieving better overall integrated performance. This is motivated by the need to obtain better reliability with increasingly complex equipment. Direct redundancy is a "brute force" approach which aggravates size, weight, and cost constraints. What is needed is a more sophisticated form of integration among avionics equipment which has more adaptive work around and back-up capabilities.

Distributed control is a technology which has great potential for providing integrated avionics systems with better flexibility and tolerance to faults. While it will generally lead to a more complex architecture, it can actually reduce hardware size and weight by eliminating the need for a central control computer.

Other advantages can be brought to avionics networks employing distributed control. "Smart" terminals can link up directly to desired interfaces without the need for centralized apriori interface definitions. This adaptive modular approach would simplify equipment interfacing, as well as government contracting, by eliminating third party interface control. Optimized interface traffic could be determined adaptively in flight.

This interface adaptability can be utilized to improve reliability with a "fail soft" capability. If secondary sources of data can be located on the bus, traffic can be rearranged in flight to support its use. One of the most powerful ways of achieving reliability without direct redundancy is to perform remote processing of crucial avionics functions. To achieve this, less critical processors would have to be reconfigured with back-up programs. These remote programs would manipulate critical function input/output (I/O) over the multiplex bus providing a degraded back-up capability. A Navy study has suggested bus terminals which can directly tie into the equipment I/O to support this concept.[3]

Needless to say, supporting I/O control over the bus would radically distort the optimum data traffic on the bus. However, an adaptive bus control could reconfigure interface traffic to support a wide variety of these back-up conditions.

## A CANDIDATE APPROACH FOR DISTRIBUTED CONTROL

To be sure, adaptive avionics processing and interface control for the avionic bus is a desirable objective. The literature abounds with distributed processing concepts in all levels of abstraction. To bring this theory and practice into better focus, one approach for adaptive distributed bus allocation is presented in this paper. It is based on recent work at Westinghouse on multi-processing. The proposed scheme requires that four basic provisions be made as described below.

### 1). Define Dynamic Bus Allocation Protocol

In anticipation of distributed processing, MIL-STD-1553A provides an option for passing bus control from one computer to another. As of this writing, the proposed MIL-STD-1553B defines a specific protocol for passing control which includes "handshaking". Consequently, the framework exists in which to develop architectures which exploit the potential advantages of distributed control, while maintaining compatibility with equipment designed to MIL-STD-1553A or B. This general approach is quite similar to that proposed in 1975 as part of the IEEE 488 bus.[4] Very little use has been made of this 488 capability to date because of the lack of distributed operating systems.

There are several more advanced approaches for distributed bus control discussed in a recent AFAL report.[5] Generally, they are a radical departure from the command response precept of 1553 and are, therefore, not considered here. However, the bus contention schemes discussed in this report should be considered on future buses. The naturally adaptive features of uncoordinated bus contention could simplify the start-up and bus allocation requirements described below.

2). Initialization

The initial coordination of an associative control scheme must be given special consideration. Since there is no one controller, there is a distinct ambiguity during system initialization. The easiest approach is to identify one processor as the initial organizer to begin the associative process. Naturally, in the event of initialization problems, back-up processors must be prepared to intercede.

3). Centralized Data Base

A central data base, such as a mass memory, would be used in place of the control computer for system bus real time control and coordination. The central data base would be composed of three types of data:

- o Bus Management Data
- o Real Time Clock
- o Back-up equipment configurations

Functionally, the central data base provides the command sequence among controllers with a real time reference. A central data base can also serve to down-load intelligence into local processing elements for fail soft capability as described above.

To minimize fault dependency on a central data base, some form of back-up capability must be included. Critical data could be replicated in the "smart" terminals to support back-up communications in the event centrallized bus management data were not available. Further, in each avionics subsystem "hard" programming of some local intelligence must be maintained to insure a minimum systems capability.

4). Associative Control Algorithms

Recent work at Westinghouse has investigated distributed control techniques for multiprocessing.[6] Many of these concepts are also applicable to decentralized multiplex bus control. In a general sense, bus time allocation is directly analogous to task scheduling in multiprocessing. A central authority can stand "off line" and direct traffic (or processing). This authoritative approach is easy to understand and visualize. However, the more efficient architecture is to have the units do their own scheduling. This leads to a more adaptive structure where the units work in association with each other.

Two distinct control approaches were studied. The first was a "one time" calculated control sequence which remains fixed until conditions change, such as a failure or mode switch. Figure 1 shows the major decisions which must go into the optimum control sequence lists. The associative controllers would jointly establish resources, interface priorities, and timing requirements under interim control. This overhead for calculating a fixed timing sequence on the bus is obviously very small. The complex optimization and

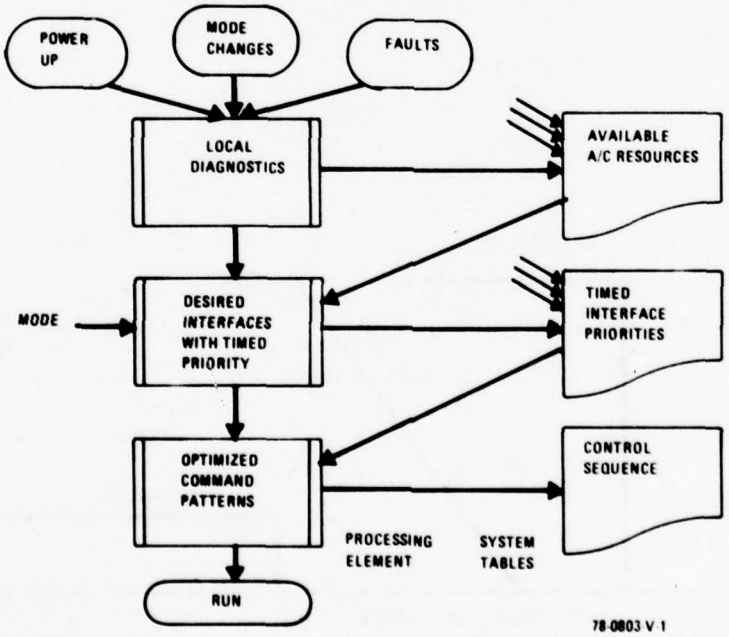



Figure 1. Calculating an Optimum Control Sequence

resource management calculations are performed only once per configuration. Thereafter, each controller merely passes control to the next on the list using the MIL-STD-1553B dyanmic bus allocation protocol after the agreed upon bus control time. If the bus traffic matches that projected by the calculations, this is a highly efficient approach. Because it is fixed, however, it degrades if the traffic varies significently from the predicted.

A second distributed control approach developed in the Westinghouse study was "Dynamic" bus allocation. Here, the optimization of prioritized bus traffic would be calculated "in-line" with each bus control sequence. Based on time dependent interface priorities, control would be passed to the next most important controller, again using the MIL-STD-1553B dynamic bus allocation protocol. Figure 2 shows how different types of interface control priority characteristics could be managed. "Hard", as well as "Soft", interface timing sequences could be managed through this distributed control. Dynamic interface optimization would, indeed, require more overhead; however, it would be much more responsive to real time bus traffic pattern changes.

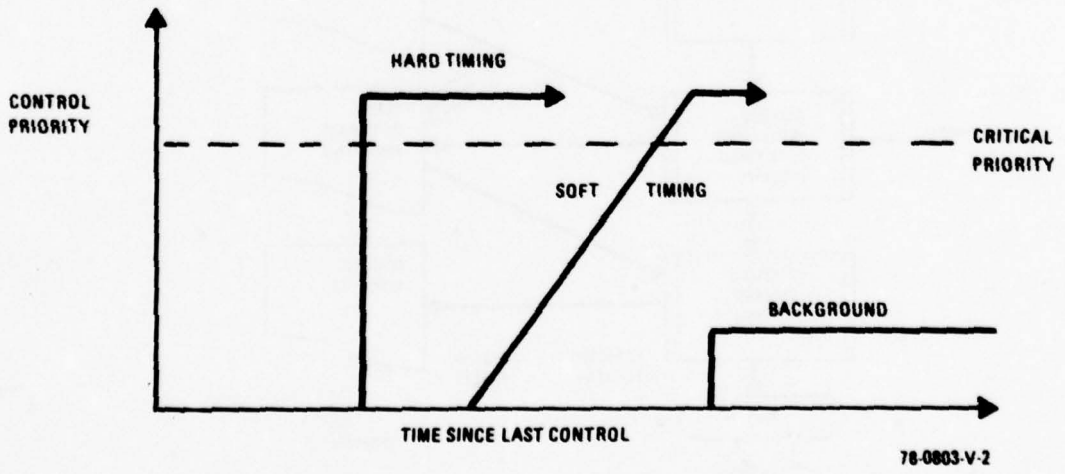



Figure 2. Timed Control Priority

## CONCLUSIONS

Future avionics needs for increasing capability and reliability argue for more sophisticated integration of equipment. The recent proposed changes in MIL-STD-1553B further defining dynamic bus control should be exploited to allow multiple "smart" terminals to control the bus. The next step is to develop adaptive distributed control approaches like the ones presented in this paper.

REFERENCES


1. Computer, August 1977 Volume 10 #10 p 120 re: Creative Stategies, Inc.

2. Aircraft Internal Time Division Command/Response Multiplex Data Bus MIL-STD-1553A, 30 April 1975.

3. Feasibility Demonstration of Distributed Processing for Small Ships Command and Control Systems, Honeywell and Naval Electronics Laboratory Center, August 1976.

4. Digital Interface for Programmable Instrumentation, IEEE Standard 488-1975, October 1975.

5. The Impact of Wideband Multiplex Concepts on Microprocessor - Based Avionic System Architectures, Honeywell and AFAL, February 1978.

6. Multiprocessing for EW Avionics, C. Wise and R. Koppelman, NAECON 78 May 1978, Dayton, Ohio.

# INTERFACES AND DATA TRANSMISSION ON A PARALLEL BUS FOR MODULAR EQUIPMENT

Dr. Walter Metzdorff and Dr. Manfred Mall

DORNIER SYSTEM GMBH, GERMANY

ABSTRACT

Functional partitioning of on-board integrated data acquisition, processing and distribution systems can have impact on cost, reliability, maintenance, and other factors.

If data processing systems are to be partitioned from the unit level (i.e. a box containing a number of modules) to the module level (e.g. multiplexers, A/D converters, memory modules ....), standard electrical interfaces and standard transfer procedures must be established between the different modules.

On the unit level the serial data bus acc. to MIL-STD-1553 forms the data highway between LRU's.

Investigations towards appropriate solutions for interfaces and transmission techniques between a large number of modules have resulted in the development of a parallel bus as the dataway. This is the best compromise between flexibility, cost,performance, complexity, and adaptability to technical innovations.

For a consequent system structure the partitioning of hardware and software is of the same importance. Since the hardware is well structured through the interface specification of the serial data highway, the parallel data way and the functional modules, an executive and high order real time language compiler has been sucsessfully applied for systematic software production.

## 1. INTRODUCTION

Higher mission requirements of aircraft weapon systems are leading continuously to increased performance of (Avionic-, Guidance and Control-) Onboard Electronic Systems. This has led to a remarkable increase in the number of electronic systems aboard the aircraft and to an extremely large increase in complexity and cost in all the considered phases (development, production and mainly in the life cycle). New structures, systems techniques and technologies under consideration have to have the potential for the required increase in mission capability and configuration flexibility with less complexity, cost, crew workload and cockpit panel space.

The traditional architecture has been an amalgamation of nearly autonomous subsystems for all different functions, each subsystem having independent sensors, processors, effectors (displays/controls). The partitioning of the new integrated system architecture is shown in Figure 1, Experimental Flight Control and Night Vision System for Combat Helicopter. An integrated sensor system generates all required physical states for all functions. Through the integrated data acquisition/processing/distribution system all states are acquired and all functions are implemented as application software. Manual inputs/outputs are provided through an integrated effector system.

A main contribution towards achieving a reduction in complexity and cost is a higher degree of commonality (standardization) between and within all LRU's of the integrated data system. MIL-STD-1553 as a standard interface and protocol for data distribution between LRU's is the step towards commonality between LRU's in a distributed system. This data high way is a tool for functional partitioning and flexible modular integration at the LRU-level.

A second, but consequent step with the goal for higher commonality is comparable to the MIL-STD-1553 approach defining a standard for the data way on the LRU level. A standardized data way on the module level within each LRU is proposed. Figure 2, Basic Data System Configuration, illustrates the concept. IC-boards as modules in each LRU representing one or more functions as I/O-Converters (ADC, SDC, FDC, ... and vice versa ... Mil-Bus-Couplers, ...), data processors and memories have to be connected. A parallel bus has proven as an appropriate solution for the interfaces and

transmission techniques as data way between a number of different or identical modules. As in the MIL-STD-1553, an interface and protocol specification is provided leaving space for the individual engineering development of functional modules for data acquisition and processing in the LRU and for technological innovations. But, for a certain time period, also a developed set of functional modules can be used in all different LRU's in a system in which the same functions are required (I/O, data processors, memories ...) with the expected implications on cost due to a higher degree of commonality.

The concept utilizes the MIL-STD-1553 data bus as the data highway for communication to and from all LRU's. All LRU's are designed out of functional modules arranged along parallel data ways.

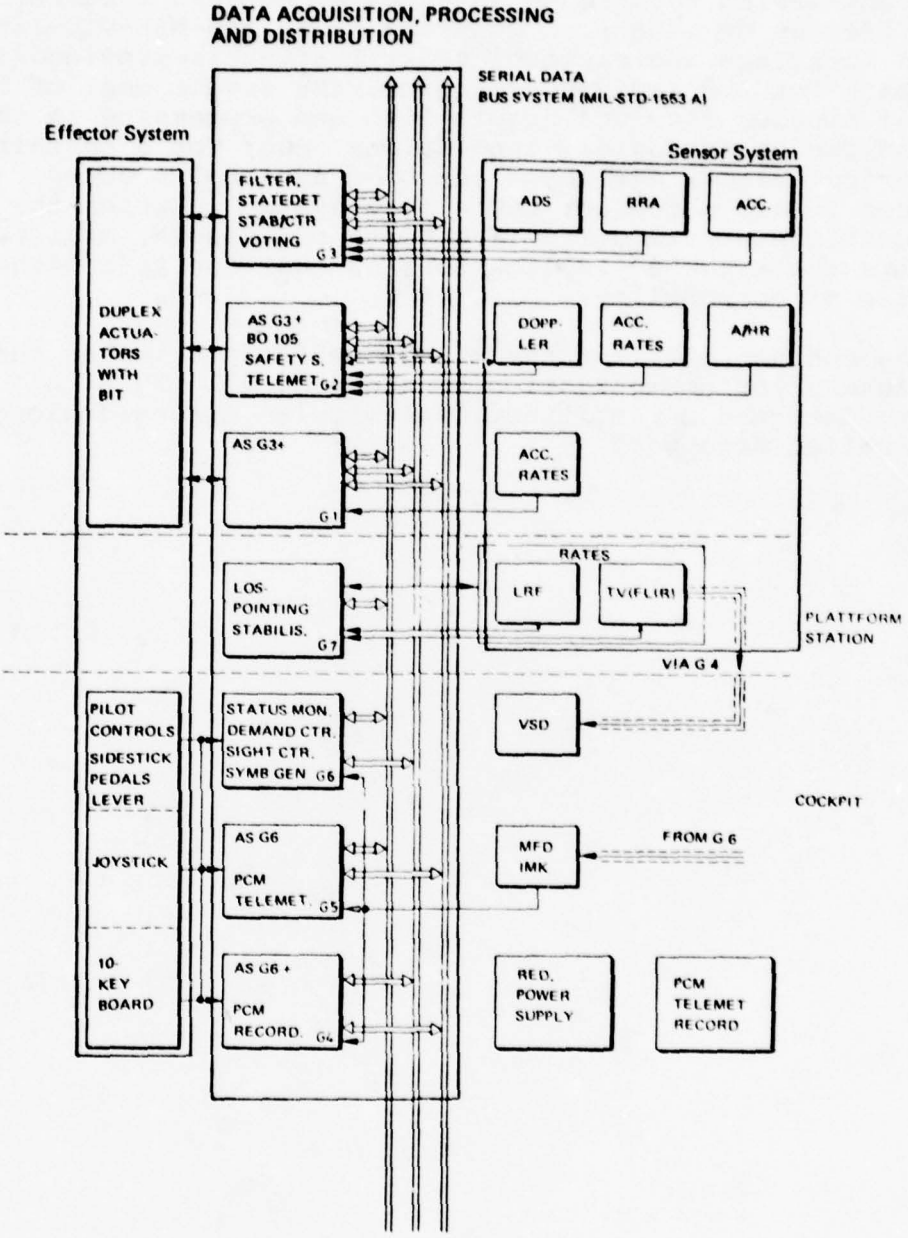



Figure 1

DORNIER

**Experimental Flight Control And Night Vision System For Antitank Helicopter**

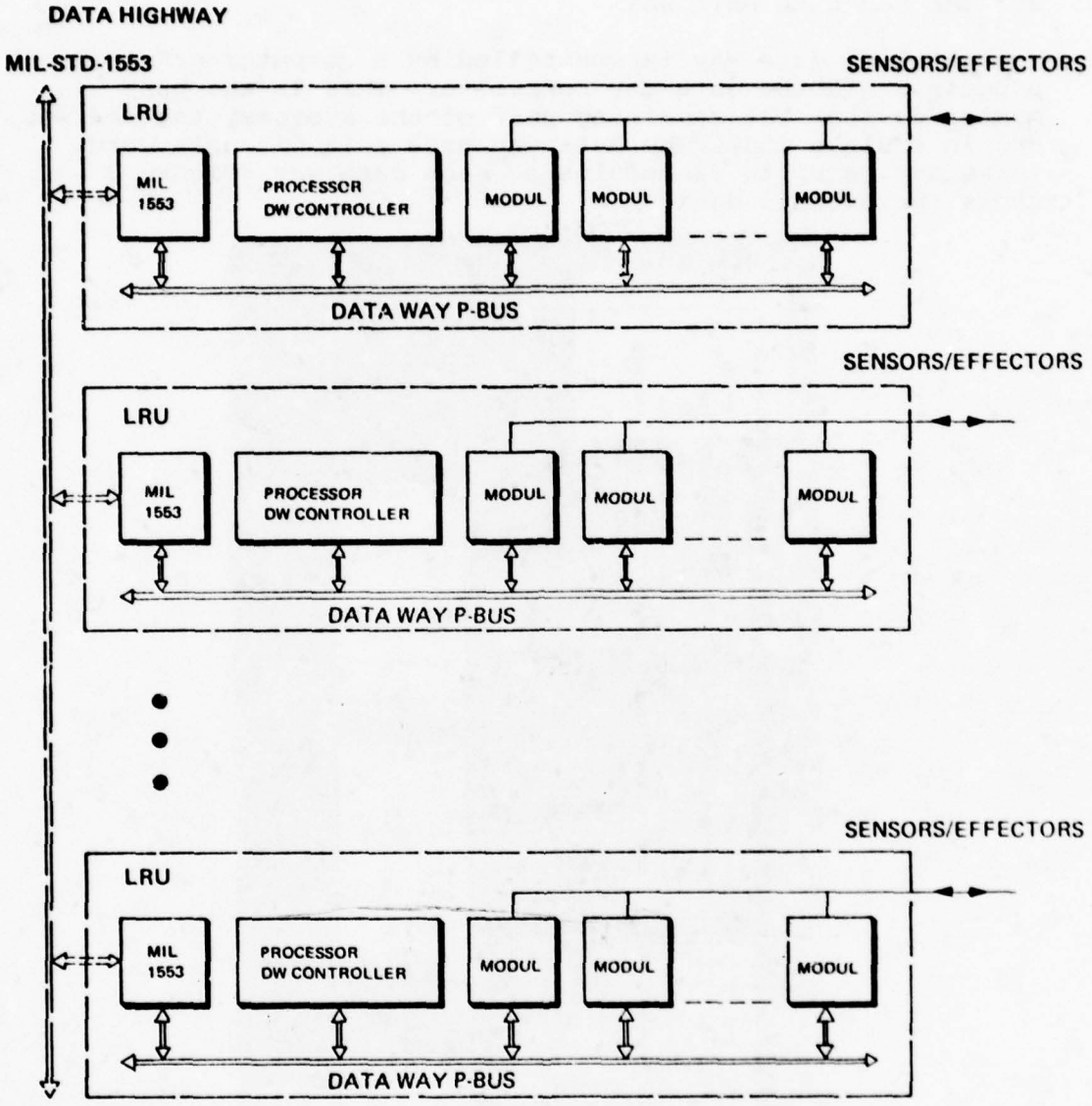



Figure 2

DORNIER

**BASIC Configuration**
**Integrated Modular Data Acquisition, Processing And Distribution**

## 2. THE PARALLEL DATA WAY

The parallel data way is controlled by a computer or microprocessor via the data way controller. This is the bus-master-device. The remaining part of the systems, the modules, are in a slave mode. The data way uses a 16 bit data word. There may be up to 32 modules on each data way. Figure 3 shows the overall design.

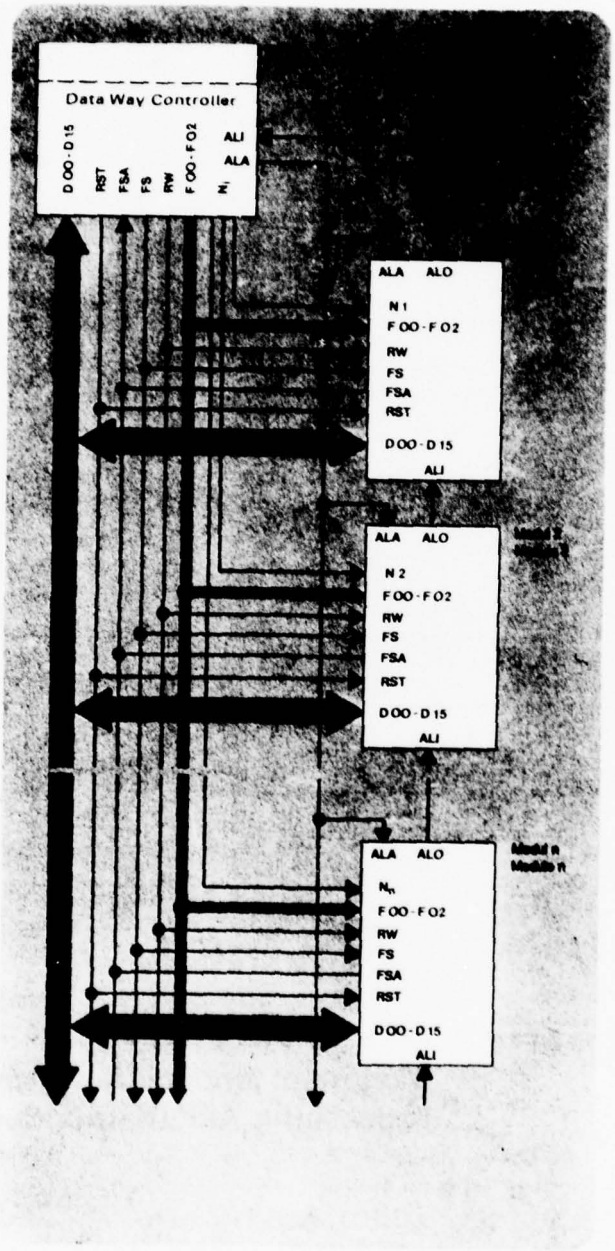



**Figure 3 : Data Way Signals and Lines**

Addressing of the modules is done by 32 stich lines N running from the data way controller to the modules. Three function lines FOO-FO2 are used to command and control the modules. This lines are decoded in the modules.

These modules are usually passive, operating in slave mode. But there are provisions for a vectored interrupt system. This uses a daisy-chain approach through lines ALI, ALA and ALO leading to one hardware priority. If there is an interrupt pending, the master device transmits a "give me the interrupt code" command and the interrupting module transmits the module address back, which is used as an interrupt pointer.

The system was designed for a very simple, easy to understand operation. This leads to only three different data way operations:

- WRITE data from the master device/data way controller to a module
- READ data from a module
- READ INTerrupt vector.

The data way is 16 bit wide using 16 parallel, bidirectional lines. Transfer is synchronized by a hand-shake protocol. Hand-shake is on both, leading and trailing edge of data and is implemented using two lines, a function strobe line FS and a function strobe acknowledge line FSA. The direction of transfer is marked by line RW (Read/Write). The manner by which a transfer is established is illustrated in Figure 4. A read transfer here is used for explanation.

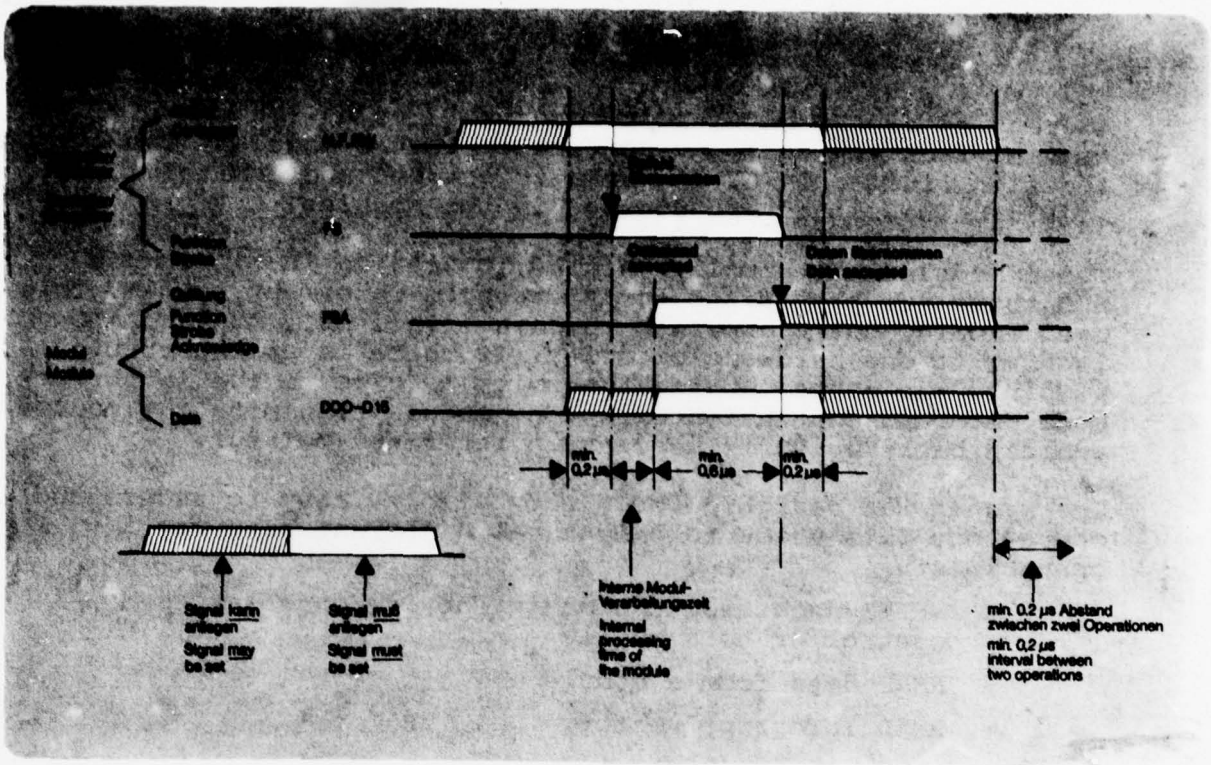



**Figure 4 : Read Transfer from Module to Controller**

Read transfer from module to controller - The controller or processor generates the address signal N and transfers the instruction (module function F and direction of transfer RW) to the dataway. The addressed module decodes the command and connects its data lines to the dataway. After a certain time, about 0.2 µs, required by the module to safely recognize the command, the controller/processor generates the function strobe "FS". This signal initializes the desired module function. If this is a valid function, the module generates the function strobe "acknowledge signal FSA". The FSA signal is not set in case of an not allowed function desired from the module. The controller/processor accepts the data transmitted by the module and resets the function strobe, signal FS. This causes the module being addressed to reset the FSA-signal. Data and command channels are maintained for approximately 0.2 µs after the FSA is reset. A short time-interval is inserted between the end of a read transfer and the beginning of the next operation, to allow for a possible interrupt to be properly recognized.

The write transfer is very similar. All operations are performed in an asynchronous mode. The timing is very simple, only the necessary minimal times are specified. Table 1 summarizes the function of all incorporated data way signals.

TABLE 1

Data Way Signals

| Section | Signal Name | Symbol | Line Characteristic | Signal Direction |
|---|---|---|---|---|
| Command | Module Address<br>Module Function | Ni(i=1...32)<br>FOO, FO1,FO2 | Dedicated Lines<br>Unidirectional Bus Line | Controller→Module<br>Controller→Modules |
| | Direction of Transfer | RW | Unidirectional Bus Line | Controller→Modules |
| | Reset | RST | Unidirectional Bus Line | Controller→Modules |
| Data | Data | DOO-D15 | Bidirectional Bus Line | Controller↔Modules |
| Control | Function Strobe | FS | Unidirectional Bus Line | Controller→Modules |
| | Function Strobe Acknowledgement | FSA | Unidirectional Bus Line | Modules→Controller |
| | Collective Alarm | ALI | Collective Line | Modules→Controller |
| | Alarm Interrogation | ALA | Unidirectional Bus Line | Controller→Modules |
| Special | | NSL1, NSL2<br>SL1, SL2 | Dedicated Lines<br>Unidirectional Bus Line | Controller→Module<br>Controller→Modules |

It should be understood, that in an interface standard, not only the data way and its protocol has to be specified. It also is necessary to specify signal levels, physical dimensions, connectors and so on. Signal levels and power supply voltages are adapted to CMOS (RCA 4000 and similar series); the physical dimensions of the modules fit in an ARINC 1/2 ATR box

The standard described here in an abbreviated form is not firmly standardizing the total system layout. But it is sufficient and leads to flexible, well structured, easy to service equipment and, as a result, to reduced life time cost. Since it is only an interface standard it accomodates future technologies and future requirements at minimum cost.

SOFTWARE

The system described herein offers a set of established rules for designing well structured hardware. This is of great influence and help within the area of software.

For many years the German Government has sponsored efforts to develop a real-time-language for embedded computer-systems in the industrial area. This language is called PEARL (Process and Experiment Automation Realtime Language) and includes all necessary features for the programming of real time systems. The level of the language is rather high as compared for instance with the DOD-1 system-language under definition now. One very important feature of PEARL is the inclusion of a "system-part", which is used to define the hardware environment of the specific software. For instance, it is possible to define the hardware down to registers in I/O-modules. The features of the language allow for low cost programming, transparent documentation and low life time costs of software. It should be understood that a language of that kind needs an executive system as a runtime environment. The executive is specific to the used hardware, since it is partly the compiler too. If the hardware is well and evenly structured, there is a good chance to implement executive and compiler in a very general way without paying to much for this generality.

This was done on the basis of the hardware standard presented here. The result was a PEARL-compiler and a very modular executive system. The generation of the specific executive is done at compile time during code-generation. The compiler uses the information given in the systems part of the program and a list of the language constructs called. The results of the effort are encouraging. For medium sized systems using several tasks, clocking, sema's etc. the size of the executive about 1 K 16 bit words. This is about 0.25 K overhead compared to a tailored solution written in assembly language.

The overhead produced by the compiler is in the order of 10% to 50% depending on the language used. The 10% overhead includes arithmetic, simple branching etc. The 50% overhead occurs as a worst case with array processing in indexed loops. The total overhead produced by the compiler - without exetutive - is usually in the range of 20%. This seems to be an acceptable result, achieved by using a well defined and structured hardware as described above.

# GENERAL PURPOSE AUDIO-MULTIPLEXED INTERCOM SYSTEM

G. R. McClure
W. R. Sobko

AiResearch Manufacturing Company
of California
(A Division of The Garrett Corporation)
*Torrance*, California

## ABSTRACT

The development of a demonstration model multiplexing system for audio intercom communications has been sponsored by the Naval Air Development Center (NADC). The model system codes audio information using a continuous, variable-slope, delta-squared modulation technique. Audio frequency response is limited for voice recognition and intelligibility, minimizing data rate requirement. Relations between audio sampling rate, intelligibility, bus loading, and capacity for additional terminals are discussed.

Transmission is in MIL-STD-1553 formats. A form of dynamic bus allocation is used to control the intermittent requirements for data transmission. The system can share a bus with other avionic systems using 1553 protocols. Bus availability requirements for a shared system are discussed, relating availability to apparent audio delay.

Advantages of the digital system are covered, including ability to share communications channels with other data, suitability for scrambling or encryption, and incorporation or alarms or warnings.

## PROBLEM STATEMENT

The problem statement is associated with the development, design, and fabrication of a five-station, general purpose audio-multiplex intercom system (ICS) based on a 1553 time division multiplexing technique for the purposes of experimentation and laboratory evaluation by NADC. The

PRECEDING PAGE BLANK

objective of this program is to provide an audio-mux system for general purpose use within Navy airborne platforms. The system must be structured so that it can be contracted or expanded to accommodate a minimum of two audio stations to a maximum of twenty. The approach to the audio-mux is to be one of total system integration and not as a separate piecemeal system. The multiplexing of audio signals must be accomplished using transmission over a MIL-STD-1553 Command/Response System and/or over the polling structure as defined by MIL-G-85013(AS).

## SYSTEM CONFIGURATION

The basic audio-mux as shown in Figure 1 consists of a number of stations (from 2 to 20), avionic tone simulator, and a system bus controller. The system uses a common bus to multiplex data and commands between stations and between the bus controller and stations. Communications protocol on the bus is per MIL-G-85013(AS) polling-contention mode, while data transfers between elements of the audio-mux are performed per MIL-STD-1553A waveform and timing requirements. This combination is equivalent to MIL-STD-1553B dynamic bus allocation.

The avionic tone simulator generates tones associated with particular avionic equipment. These tones are steered by the switching matrix to various stations to simulate several warning tone sources. Each station receives inputs from the avionic tone simulator switching matrix, microphone, and the intercom system mux bus, and provides outputs to the mux bus, headset, and recorder.

## SYSTEM FEATURES

The AiResearch general purpose audio-mux system is designed for five stations, expandable to twenty or more. The communication method and transmission format adopts the physical operating and general message characteristics associated with MIL-STD-1553A (or B). All message protocol and associated control data and status words are transmitted over the same 1553 bus.

Multiple communications modes highlight yet another feature of the ICS. This system is designed to provide the following operational modes:

(1) "Party line" operation--one person talks and all others listen.

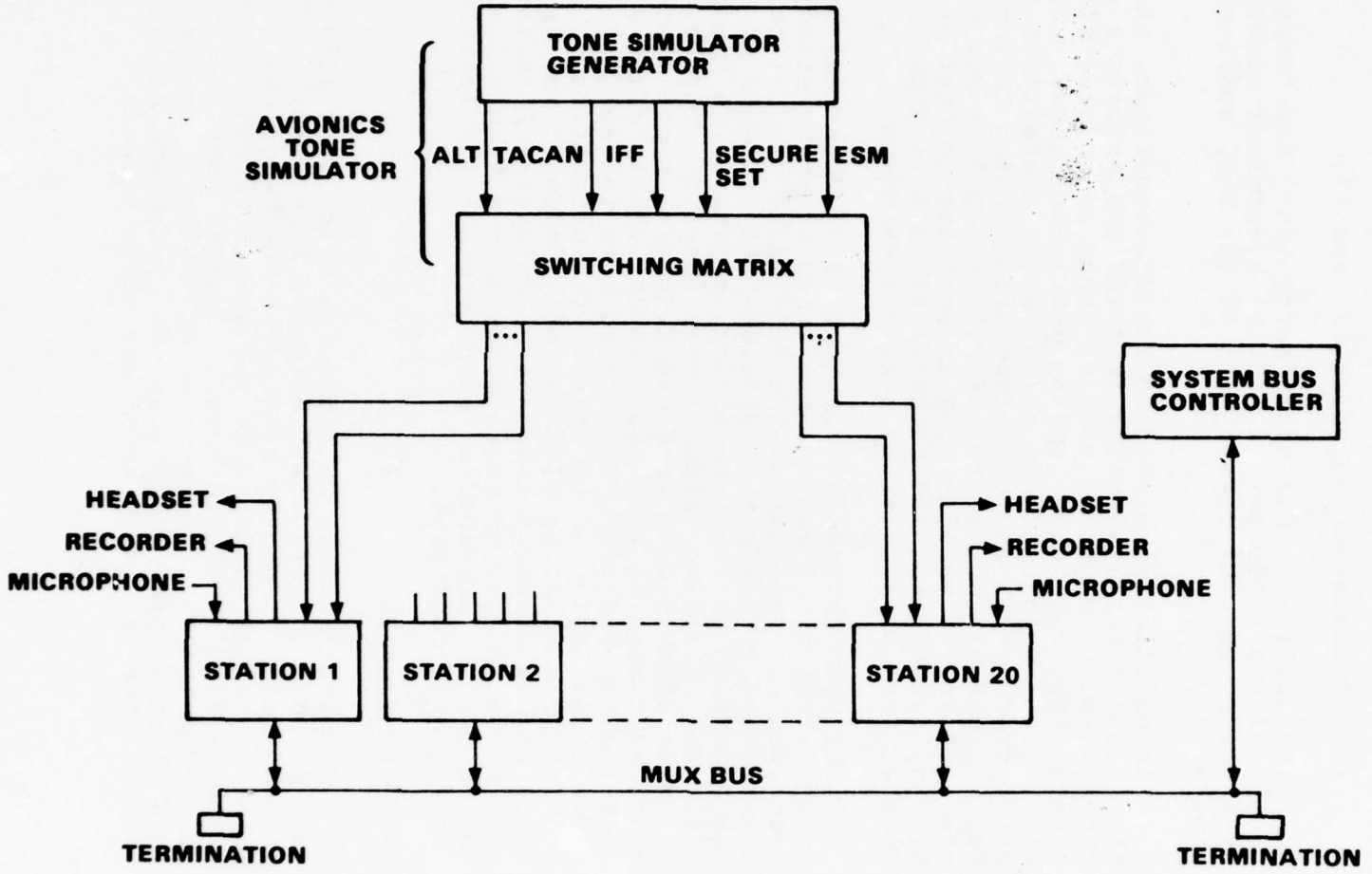



Figure 1. Audio-Mux Block Diagram

(2) "Separate" conversation--multiple 2-way communications. Bus capacity up to approximately ten conversations.

(3) Combination of modes (1) and (2).

The ability to mix both "party line" and "separate" conversations represents a natural by-product associated with the successful development of the NADC audio-mux program.

In addition to providing for conversations, the ICS also handles aural warnings from various avionic sources. These warnings might consist of those associated with IFF, sonar, RAWS, altitude, landing gear, and others. Generally, these warnings have unique audio characteristics that allow segregation as to the specific warning taking place. For example, an oscillator of 1050 $\pm$50 pulses per second produces an IFF warning tone in contrast to that of summing 512-Hz, 614-Hz, and 768-Hz oscillators that provide still another tone particular to altitude warnings.

Any warning tone or combination of tones may be switched to any station. These warning tones are then encoded to a specified bit position of the first data word of the message transmitted. This method provides the presence of this warning condition to other receiving terminals via the multiplex bus. The receiving terminal then reads the corresponding bit position of this data word and triggers an oscillator. This produces a warning tone that can be audibly mixed with an on-going conversation, permitting both the warning tone and voice communcation to be heard. The demonstration system handles five warnings with capacity designed for sixteen, requiring no change to message or data word formats. Since multiple warnings may be introduced to a terminal at one time, indicators are provided at each terminal for verification of each warning condition.

AUDIO PROCESSING

In past reviews, multiplexing voice over 1553 type data bus has appeared difficult and impractical. This stemmed mainly from the continuous nature of the analog voice signal as compared to the quantized 1553 digital system's 20-microsecond words. Systems have been considered where the transmission of a single voice signal, in a digital form, would load the system and not allow time for the transmission of any other signals.

A different system concept from those previously considered, along with modern integrated circuits, have made audio-multiplexing practical for future systems.

A first consideration in digital audio-multiplexing is how much data, bits per second, is required to be transmitted. The AiResearch-developed entertainment multiplexing system of a few years ago used 300,000 bits per second. This was, in fact, a hi-fi system. Some of the new companding (nonlinear PCM) integrated circuits for telephones use 8 kHz samples of 8 bits, 64,000 bits per second. These systems should produce extremely high voice quality. Recently produced integrated circuits for voice use 16,000 bps and greater bit rates. These units use versions of delta modulation (one bit A/D) called continuous, variable-slope delta modulator and accommodate a wide dynamic range of signals. Aural warning systems have been built (AiResearch for NASA) using delta modulation at an even lower data rate (10,000 bits/sec), but voice quality was just adequate for the intended word recognition requirement, but not for voice recognition.

The 1553 data bus, of course, has a capacity of 1,000,000 bits per second. The NADC specification called for a system expandable to 20 stations. If all 20 are allowed to talk at once (not necessarily a requirement), then, not considering bus operating overhead, the maximum sample rate would be 50,000 bits per second. Synchronization and parity bits associated with 1553 transmission use 20 percent of the bus capacity, lowering the maximum possible rate to 40,000 bits per second. Other overhead associated with the bus operation reduces the bit rate even further. The audio-multiplexer uses a 25 kHz sample rate to meet the 20-station requirement with reasonable quality. Voice recognition at this rate is very good. Future systems may use lower frequencies (like 16 kHz) with newer integrated circuits.

At 25 kHz sample rate, a bit is generated every 40 microseconds. Since this is two 1553 word times, it is impractical to handle the bits individually over the bus. However, if bits are stored in the transmitting station until an appropriate quantity is available and then transmitted in groups, a convenient system may be devised.

Consider the following analogy: there are two large buckets, as shown in Figure 2. One bucket has water running into it at a fixed rate, and the second bucket, with an outflow spigot, has water running out at the same fixed rate as the water to the input bucket. You stand by the buckets with a dipper. When the water level in the first bucket reaches a certain level, you move a dipper of water from the first bucket to the second. You repeat this process at such a rate that the first bucket never overflows and that there is always enough water in that bucket for your dipper when you dip. If the input to the first bucket and the output from the second are exactly equal, then the output from the second will be continuous and uninterrupted. Note that over a long period of time the rate of water moved by the dipper is (and must be) the same as the input/output on the bus. Another operational analogy to be garnered from this model: the output of a particular segment of water does not occur directly after its input; there is a built-in delay.

## ICS COMMUNICATION

The AiResearch audio-multiplex system operates much as the model described. The transmitter converts voice to bits and stores 512 before transmission can start. Approximately every 10.24 milliseconds, 256 voice bits (16 data words) are transmitted. In the receiving station(s), 512 bits are received (two transmissions) and stored before the analog output is started. The delay through the system from voice in to voice out is approximately 40 to 50 milliseconds, below the threshold of perceptibility to the user.

Upon power turn-on, the bus controller offers the mux bus to the first audio-mux station and sets an internal timer within the bus controller. The bus controller then tests the bus status to determine if the bus is busy. When the bus enters the idle state, the bus controller can offer the bus to the next audio-mux station. The bus controller continues this activity until all stations have been offered the bus. The controller then waits until the sample time interval (10 milliseconds) has been reached. The bus controller then resets the counter and offers the bus again to audio-mux system stations. When the bus is offered to a station it will not respond if it has no message to transmit, and the bus controller will reacquire the bus. If the station has a message, it transmits a command (receive) to the first station as defined by the address switches. This latter station (receiving) will transmit a status word back to the first (transmitting) station. A bit in the status

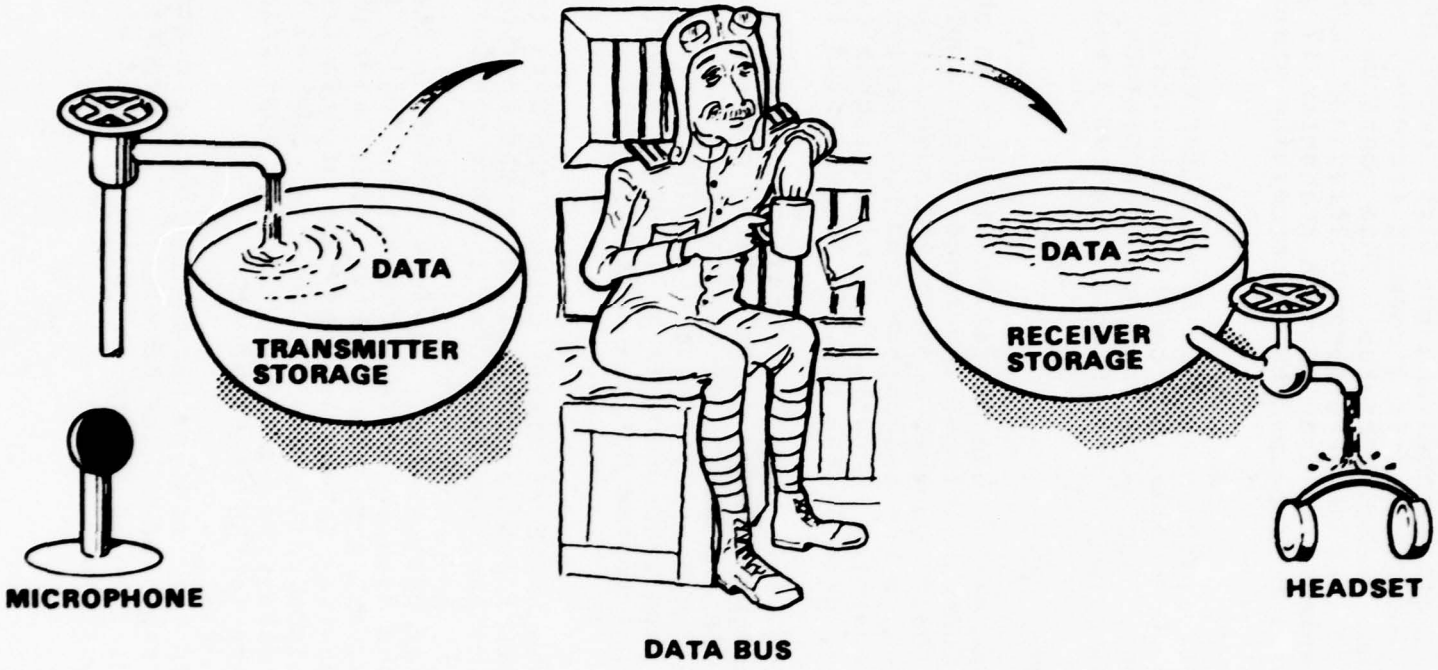



Figure 2. Bucket by Bucket Analogy

word will indicate if the receiving station is busy. If the transmitting station is communicating to other stations, it will, after receiving the first receiving station's status word, transmit a command (receive) to the next station and receive a status. This continues until all stations to receive have been notified and have responded; the transmitting station then transmits 17 data words that will be received by all receiving stations at the same time.

All stations receive a bus offer each 10.24 millisecond frame. Communications can occur simultaneously between two stations. Audio-mux communication protocol and message word formats are presented in Figures 3 and 4, respectively.

## STATION DESIGN

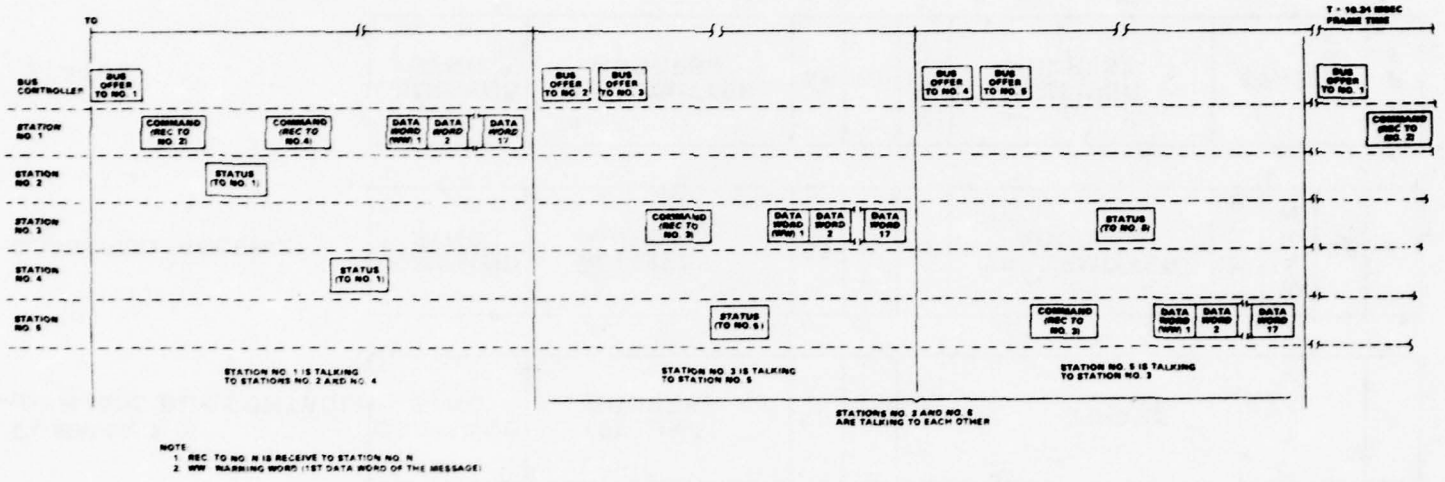
One of the demonstration stations is shown is Figure 5. A station consists of input and output amplifiers, variable-slope, delta-squared decoder and coder, input and output memories, a bus transmitter and bus receiver, switch address logic, warning circuits, and microprogram controller. A block diagram of the station is presented in Figure 6.

The input amplifier scales and filters the incoming audio signals. The delta squared coder converts the analog signal into a series of digital pulses. These pulses are stored temporarily in the transmitter memory. When sufficient data is stored, it will, under microprogram control, be output to the bus transmitter.

The bus transmitter outputs the data to the bus, again under microprogram control, when the appropriate control word has been received from the bus controller.

The bus receiver receives data from the bus and outputs the data to address comparison logic, the receiver memory, and warning circuits. Under microprogram control, the receiver determines the type of data from the address comparison units, and bus receiver control bits (command sync, data sync, etc.); then, when appropriate, it directs the data to the receiver memory.

In the receiver memory the data are stored temporarily until required by the decoder. The decoder converts the digital data back into an analog signal, which is then output to the headset.




NOTE:
1. REC TO NO. N IS RECEIVE TO STATION NO. N
2. WW (WARNING WORD) (1ST DATA WORD OF THE MESSAGE)

Figure 3. ICS Communications Protocol

S-30963

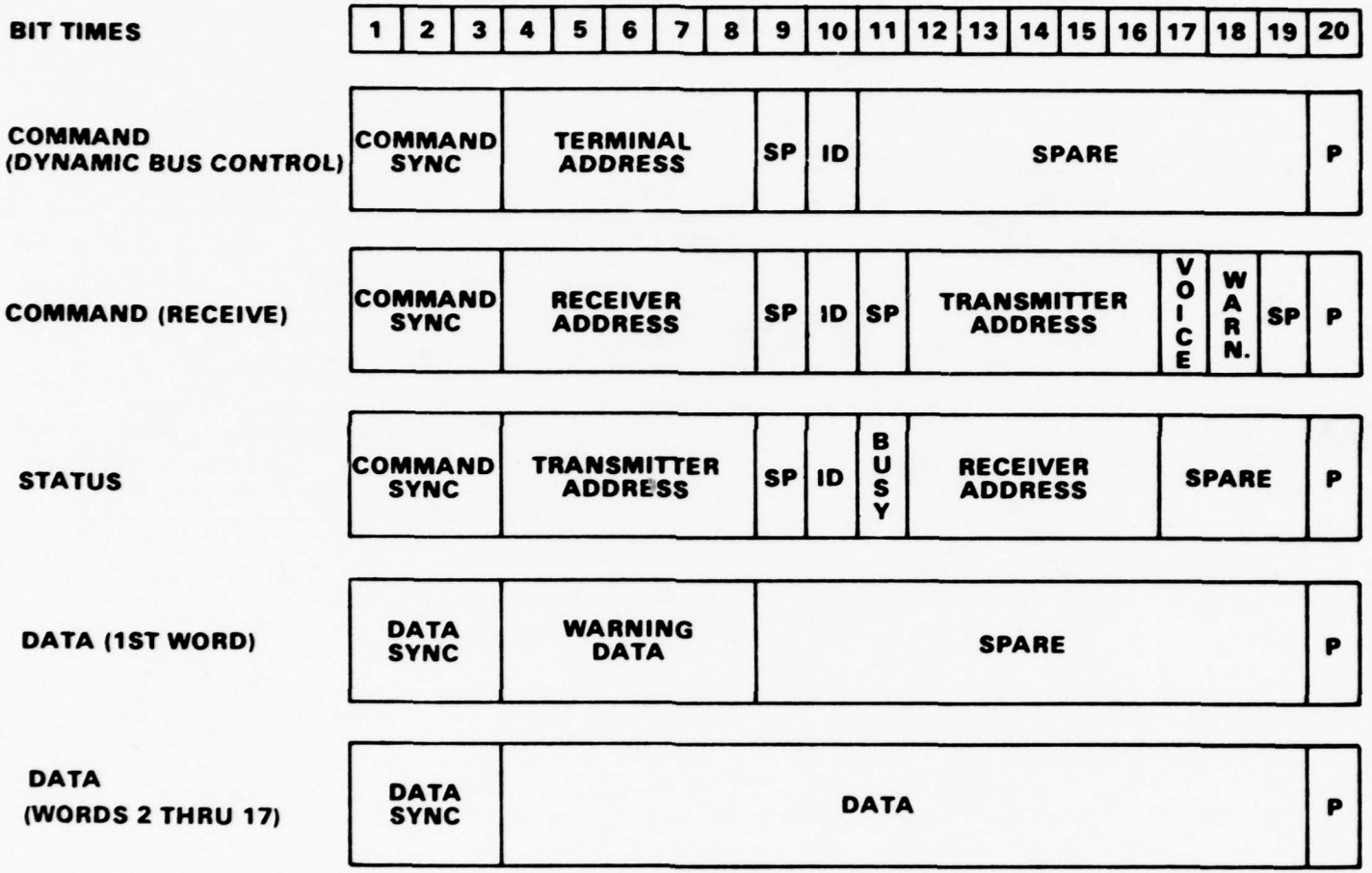



Figure 4. Audio-Mux ICS Word Formats

S-30956

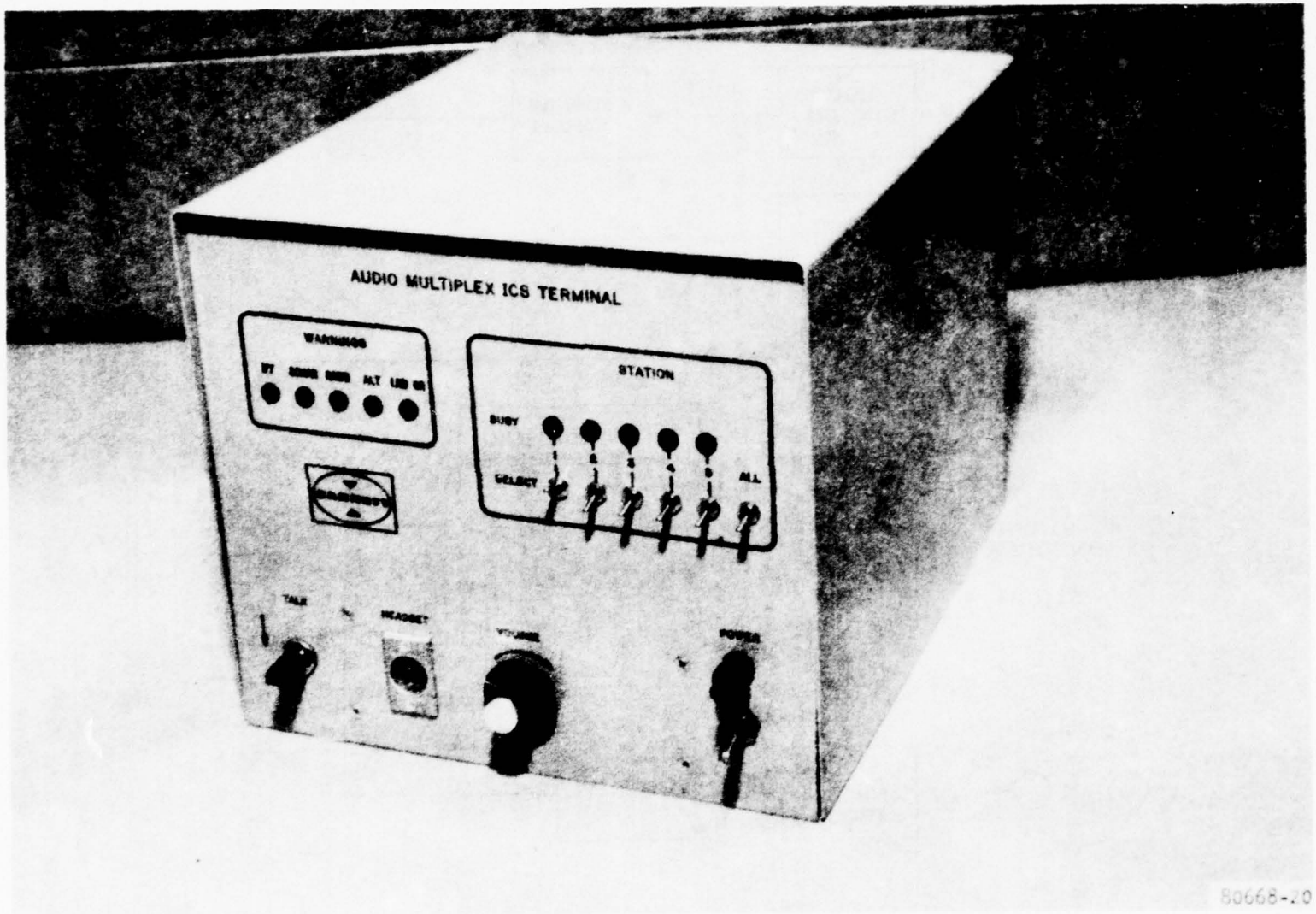



Figure 5. Typical Station

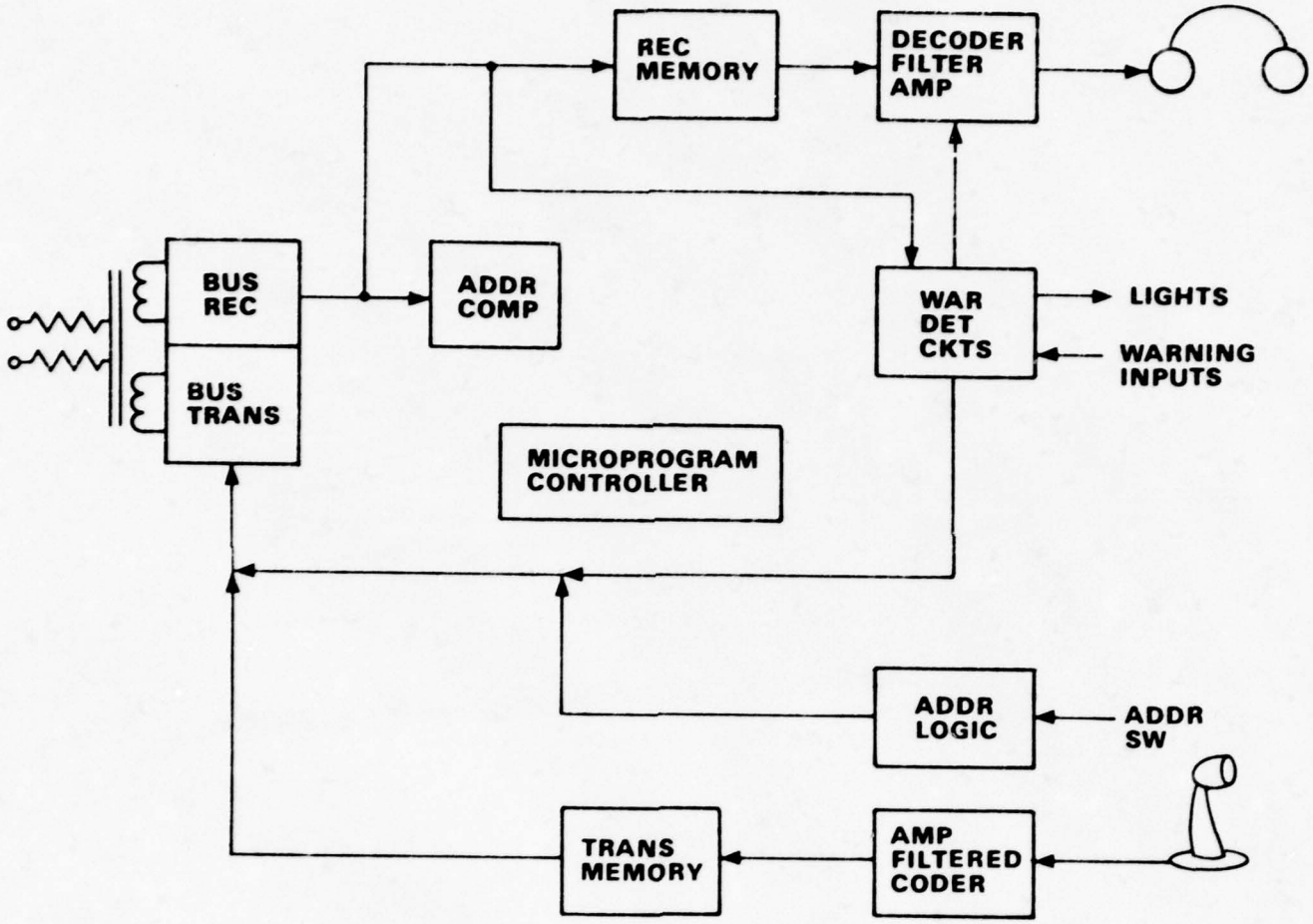



Figure 6. Audio-Mux Station Block Diagram

The warning circuits are multifunctional since they:

(1) Accept warning inputs and sum them with the audio output of the station.

(2) Set a warning discrete that is transmitted to all other stations.

(3) Accept warning discretes from other stations and start the appropriate warning oscillator.

(4) Sum the local oscillator signal into the station output.

(5) Turn on appropriate warning lights.

The switch address logic generates the appropriate station address from the toggle switch inputs. This information is transmitted under microprogram control.

Two principal areas associated with audio-mux design required particular attention during design and development phases of the system. The first was optimizing encoding/decoding for the highest degree of audio fidelity versus the number of bits per data word required to produce the audio tone. The second was the packing and unpacking of audio data which required a tradeoff analysis between message size and fidelity versus the number of stations in the system. A significant factor in this decision is the optimization of memory sizes within the station compared with transmission time and bus loading associated with the number of stations.

## FUTURE TRENDS

Many applications exist for future versions of the basic audio-mux as described in this paper. The ICS could assume a wide variety of configurations for both airborne and ground based systems. A possible configuration of the ICS for airborne use is presented in Figure 7.

In addition to the many different 1553 system configurations that the ICS might assume, it could be thought of as a prime candidate for the new technology of fiber-optic multiplexing. Due to the bus transmission rates attainable with a fiberoptic bus (around 20 MHz) and the limitations apparent in the number of stubs, audio-multiplexing represents one of many systems that could easily be integrated using this multiplexing technique.

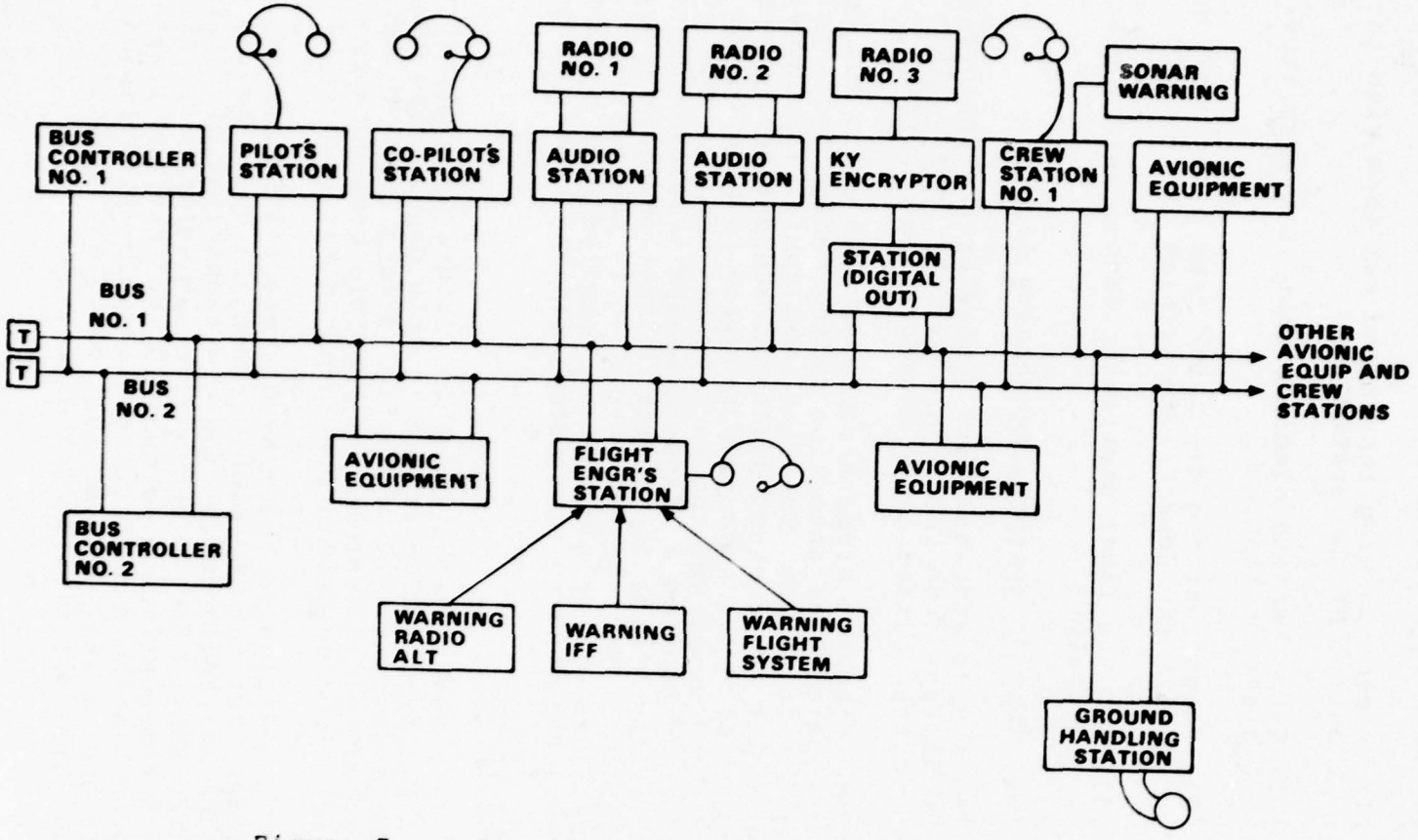



Figure 7. A Possible Future Configuration

S 309939

## SUMMARY AND CONCLUSION

The challenge of developing a practical audio-multiplexing system using MIL-STD-1553 formats is addressed by AiResearch in the design, development, fabrication, and test of the ICS demonstration system for the Naval Air Development Center. AiResearch defined a system configuration, developed the requirements, and resolved audio processing and communication tradeoffs. AiResearch also produced operational development hardware that provides a baseline for future ICS configurations.

The AiResearch audio-multiplexing system represents the first step to development of operational audio multiplexing systems. These would take into account the advantages in reducing station size, weight, and power requirements by using LSI and hybrid devices.

# DATA BUS SYSTEMS FOR DISCRETE SIGNALS

Horst Kister
VDO Luftfartgeräte Werk, Germany

PRECEDING PAGE BLANK

TABLE OF CONTENTS


1. Introduction
2. Discrete-Bus System Demands
2.1 High Reliability
2.2 Extreme Small Amount of Hardware
2.3 Decentralized Structure
2.4 Unproblematic Transmission Line
2.5 Transmission Speed
2.6 Type of Transmission
2.7 Easy and Reliable System Testing
2.8 Logical Functions
3. Basic Thoughts for System Realization
4. Functional Description
4.1 Transmission Channel
4.2 Synchronization and Addressing
4.3 Receiver Address
4.4 Fail-Safe Behavior
4.5 Decoupling of the Transmitter Stage
4.6 Logical Functions
4.6.1 OR-Function
4.6.2 AND-Function
4.7 Special Function ON and OFF
4.8 Alarm
4.9 Time Sequencing
5. System Monitoring
6. Functional Test of the System
7. Maintainability
8. Redundant System
9. Transmission of Analog Values
10. Figure 12
11. Conclusions

Appendix

1. Introduction

Modern generation aircraft are nowadays widely utilizing the Party-Line-Principle seriel Data Bus for Data Transmission between the verious "Black Boxes".

In the military domain MIL-STD-1553A is almost exclusively applied while in the civil aircraft industry a similar system, which corresponds to ARINC 429, is used.

Besides the transmission of Data between digitalized system units a more or less greater number of discrete on/off-signals is necessary between the single units or components of the aircraft.

Of course, these signals could be transmitted via a Data Bus according to MIL-STD-1553. For example a corresponding Data-Bus-Terminal would be able to process 16 discrete signals in one transmission. However as in the past, these signals must then be transmitted to the corresponding signal-source or sink via signal lines (refer to figure 1).

The following reasons are in contrast to this solution:

a. a large number of single wires must be attached to the Data Bus. System modifications would therefore require a wiring change.

b. A Data-Bus-Terminal is rather complex. The extreme demands to discrete signals with regards to reliability can only be met by employing pedrundant systems.

The above mentioned reasons should therefore justify a special, if possible very simple Bus system for transmitting these discrete signals.

This does however not preclude the fact, that signals, which are anyway necessary on the Data-Bus or which come from there, will be transmitted there. The distribution of the signals from the Data-Bus-Terminal can now be accomplished via the Discrete-Bus. Also, the majority of the signal source can now be directly transmitted to the sink via the Discrete-Bus (refer to figure 2).

2. Discrete-Bus system demands
Following are the most important requirements :

2.1 High Reliability
The reliability can be measured against the conventional, uncomplicated single wire connections. In order to obtain such high reliability values redundancy must be easily possible.

2.2 Extreme small amount of Hardware
Components generating or receiving discret signals (switches, relais aso) are usually small in size. A requirement for bigger Hardware-similar to the one used in the Data-Bus-Terminal would be inadequad. The hardware must therefore be so small in size that it can be integrated into the components without substantially changing the size or shape of the components which are commonly in use today.

Hence, the necessity to employ only one IC with the minimum amount of peripheral components is only too obvious.

2.3 Decentralized Structure

A centralized system control or system organization will only create reliability problems. Additionally, it should be possible to operate the system efficiently either with few or many members connected to it. Of course, the decentralized control of the Data flow does not exclude the possibility to channel the Data flow via a central processing unit due to system-requirements.

2.4 Unproblematic transmission line

In order to simplify the installation it should be no problem to use stub lines. Additionally the safety margin against interruptions can be increased by closing the transmission line to a loop.

2.5 Transmission speed

In order to increase the EMC-supression and to simplify the handling characteristics the transmission speed should be as low as possible but sufficiently high to cover the full application range.

2.6 Type of Transmission
Storing instructions in the receiving portion of the system increases - EM - susceptibility. For this reason, the instructions should be transmitted periodically. Also, the "ON/OFF" signals must be actively transmitted in order to eliminate any malfunction during fault - conditions or during absence of instructions when the system is only partially operated.

2.7 Easy and reliable system testing
It should be possible to monitor the system functions during operation from a central unit as well as detecting and pin-point faults to the single member.

2.8 Logical Functions
No additional hardware should be necessary to obtain the logical functions which are realized in today's systems by connecting contacts in series or parallel.

3. Basic thoughts for system realization
The full integration into the controlling or controlled system components is necessary so that the transmission-system does not appear as separate equipment. This can only be achieved if the complex functions of the system are integrated on a special MOS-chip without the need for a bigger amount of additional discrete components.

For this reason the quartz, which are normally necessary for the clock, has to be substituted by a simple RC-oszillator with a lower accuracy. That's possible using a special synchronisation-methode. So only an external capacitor is necessary for coupling the member from the Bus in case of failure. An external resistor is used, thus, due to the extremely simple construction it is possible to trible the system without problems for component size and cost. Data transmission control can be completely decentralized because of the selected type of the transmission channel (wired-or) and by simply counting, starting with a synchronization pulse . Thus, besides a gracefull degradation a system with even a few members can be efficiently employed.

The "wired or" structure of the data channel allowes the or-function without additional measures, while the AND-function can be easily accomplished by the addition of an input at the individual member. Together with the design of latches at the signal output enables the system to carry out all functions which have previously been accomplished by latch-relays.

In order to ensure sufficient noise imunity of the rather simple transmission line the lowest possible transmission speed and multiple data transmission technique was choosen instead of special data securing measures.

Only the consecutively third identical command triggers the execution of the function. In addition systems which are multiplied are working completely asynchronous. Fault pulses will therefore rarely influence the same signal on all channels, so an additional safety margin is guaranteed.

The system characteristics are rounded off by the possibility to transmit analog values of medium accuracy. This can be of advantage for a system which is controlled by a discrete-Bus if measuring data for monitoring purposes are gathered at places on which a discrete-bus data pick-off is anyway installed; for instance as with the electric power supply system (RPC, remote power control). It is, however,not intended to explain this operating mode in detail.

The example of a member for transmitting two command signals (ON and OFF) as described herein can naturally be expanded to more channels per member.

4. Functional description

The first system lay-out incorporates 256 members and the frame-time is set to 100 msec. Considering three consecutive identical command signal transmissions the total processing time is 300 msec. In most cases this should be sufficient, otherwise the alarm signal transmission, as described in a later paragraph, should be used.

Since the data is repeated at periodic intervals it is only too logical to adress via sequencing.

Using the direct adress method would only increase the amount of information without resulting in substantial advantages.

Following is a detailed description of the operation :

4.1 Transmission channel

A two-conductor shielded cable is used for the data-bus transmission line. One wire represents the bus-line and the second wire is used as a power supply line for members which cannot draw power from other sources (for instance a simple switch acting as a command originator). But one has to be aware that a breakdown of this power supply will anable all members connected to it.

The bus-configuration is like a wired or. (refer to figure 4). The two pull-up resistors R are terminating the line at the beginning and the end . In conjunction with the decoupling resistors as described in paragraph 4.6 the line capacity constitutes a low-pass. Due to the resulting big rise and fall times the line must not be terminated by characteristic impedance. It is permissible to use stub lines and the bus-line may be formed to a closed loop. The bus-line length can be up to 200 meters.

Note :
Of course it would be desirable to operate the bus-line symmetrically and inductively coupled. But at this time this does not seem to be feasable because of the resulting increase in number and size of the system components. However it can be advantagous to optically decouple the output or input signals of a member from the operated component. This can be relatively easy accomplished since the most simple event is one input and one output signal only.

4.2 Synchronization and adressing

Figure 3 shows the typical frame build-up. It starts with the frame-synch pulse. The pulse can be easily detected from the rest of the signals by its characteristic width. The end of the frame-synch pulse is a reference point for all members. Each member starts to count the time slots from 1 to N starting at this point. At the same time the time slots 1 to N represent the source address.

Each time slot is terminated by the auxiliary-synchronizing pulse, the slot-synch. The end of this pulse is at the same time the beginning of the next time-slot.

A time-slot is divided into 4 bits (a,b,c,d). Bit A is always hight because the end of the frame-synch pulse or the slot-synch pulse is the reference point.

Bit B is high when an "OFF" command is transmitted (figure 3, address 2), Bit C is high when an "ON" signal is transmitted (figure 3, address 1). It is of course possible that bit A and B is high (figure 3, address n-1). For clarification refer to paragraph 4.8.

The synchronization pulses are generated simultaneously by all members. Because of the wired-or structure of the channel the slowest member always determines the end of the synchronization pulses. Therefore, the slowest member also controls the system timing.

Without the slot-synch pulse it would be necessary that all members synchronously sweep all 256 time-slots with sufficient accuracy. But with the slot synch that's only one time slot. Therefore it is possible to employ a relatively inaccurate RC-oscillator.

At the end of the frame-time all members simultaneously generate another frame-sync pulse and the cycle repeats again (for exceptions see paragraphs 4.8 and 4.9).

4.3 Receiver address

The consecutive numbering of the time slots corresponds with source address as already mentioned in paragraph 4.2 . Hence the receiving member receives the information because it is programmed to the same time slot.

Each IC houses a transmitter and a receiver. Therefore, in order to program the transmitter address 8 Pins are required and for the receiver address another 8 Pins are also required. Taking into consideration that in the majority of the applications the execution of commands must be supervised, i.e. the received command has to trigger a feed-back signal. So a fixed assignment between command-address and feed-back-address is usefull. According to figure 5 this assignment is such that the command-address and the feed-back-address are located adjacent to each other. Therefore, only 8 Pins are required for programming, while a 9th Pin determines the function "receive and transmit feed-back signal" or "transmit and receive feed-back signal". In the case that no feed-back signal is required the corresponding feed-back address will not be used.

4.4 Fail-Safe behavior

During the partial operation of the system or during fault conditions the transmitting function of a member can be impaired.

The following can be a failsafe position of the receiver :

a. no command-signal→"OFF"

b. no command-signal→"ON"

c. no command-signal→"maintain the last valid condition"

All three modes mentioned can be programmed into the IC.

4.5 Decoupling of the transmitter stage
According to figure 6 the decoupling resistor Re is used to avoid a break-down of the remaining system in case of a shorted transmitter stage. Additionally, this makes it possible to localize the disabled member. Therefore, in case of a no malfunction condition a signal voltage is present on the Data Bus per figure 7a. One shorted output transistor effects the signal per figure 7b, two shorted output transistors effect, figure 7c etc.

Thus, the receiver must be able to exactly evaluate the signal despite a change in amplitude and BIAS-level. This is accomplished by employing the technique per figure 8.
The peak-value of the signal voltage is stored in C1. Divider R1/R2 produces a voltage which is equivalent to 75 % of used as reference for the receiver-comparator COMP 1.

The switching-level is therefore always in the middle between the upper and the lower signal level. This assures the proper functioning of the system up to three shorted transmitter stages. However due to the lower signal level a higher fault EM-sensitivity must be taken into consideration.

## 4.6 Logical Functions

### 4.6.1 OR-Function

If members A and B are programmed to the same address per figure 9 the result will be an OR-signal due to the wired-or-structure of the Bus. This signal is then received by member C.

It is of course possible to program any amount of members to the same address.

### 4.6.2 AND-Function

Each member has its own additional AND-gate. As per figure 10 member A transmits its information to member B. Within the integrated AND-gate this signal is then tied together with the input signal E2. The resulting AND-signal in turn is transmitted to member C via the adjacent address.

Of course, any number of member can be tied into AND-functions.

It must however taken into consideration that the E2 is not available on the data-bus for possible further use (for clarification refer to paragraph 7).

4.7 Special function ON and OFF

As described in paragraph 4.2 the bit combination b and c can be both high. A specific usage for this mode has not yet been assigned. It could be thinkable to use this combination to initiate test functions within the receiver. For example the signal E2 could be connected directly to the bus for test purposes in order to localize a possible fault within complex AND-functions.

4.8 Alarm

If an shorter reaction time is required between several members, then the corresponding transmitter can arrange a priority signal transmission by prematurely triggering the frame-synch pulse for three consecutive times in case of a signal change at the input. Of course all members which are entitled to execute an alarm-transmission will be programmed to the beginning of the time sequence.

4.9 Time Sequencing

If the system uses a considerable smaller number of the scheduled 256 members then one member- preferably the supervisor-can interrupt the time sequencing at any one point by generating a frame-synch pulse and start the sequence again. This will shorten the reaction time of the system.

This function must not be executed simultaneously by all members because in case of failure the system will continue to operate, however at a somewhat slower speed.

5. System monitoring

A special system member-the supervisor- will fulfil this task. Like any other member it is connected to the data bus (refer to figure 11). It has monitoring functions exclusively. It monitors the presence of command signals, the concurrence between command signal and feed-back signal and the concurrence of the single channels in redundant systems. In case of a fault it shows the address of the involved member(s) and channel ,therefore it is possible to automatically localize faults even to the single member.

Fault detection and fault reporting is a central function. In case of failure of the supervisor the overall functioning of the system is not impaired.

6. Functional test of the System

The initially inactive supervisor can be supplemented in order to test the functioning of the system (for instance preflight check). According to a special test program stimuli signals are generated and the system reaction tested as previously described. These stimuli signals can superimpose the original signals. A high or a low signal can be applied to the bus, with low impedance, so overriding a members signal.The inflight testing of the transmission system can be achieved by periodical injecting stimuli signals and cross-examination the systems response.

7. Maintainability

The described supervisor will allow very easy maintaining the system because it is pin pointing a fault down to the single member.

But if there is no supervisor for instance if a system is checked out in steps starting with only some members a very simple little test-book could be usefull. This littel test-book would be connected to the bus using test-socket at convience places. With a digiswitch a source-address can be selected and the signal indicated. The same way a source address can be selected and a command impressed to the bus. So it's not only possible to check the discrete bus system but check also the controlling or controlled system componente.

It is obvious that in the same way a fully automatic text system could do a complete system check.

8. Redundant System
I fail operational system is a must a bus system can be easily multiplied. Because the system is a extremely small and cheep there are no cost or space limitations. The systems will run fully independent and not synchronized. Due to this a noise pulse will hardly influence the same address so improving the EMC characteristic.

9. Transmission of analog values
For this function a slightly modified member is used. One pair of addresses is carrying 8 data-bit. This resolution normally is sufficient for supervisory purposes. A functional description of this member will be given later.

10. Figure 12 is showing the block-diagram of a receiving and transmitting member.
Having in mind the earlier paragraphs it should be self explaining.

11. Conclutions
In our opinion the main advantages of the proposed systems are :

a) gracefull degradation due to the decentral controlling method

b) extremely simple hardware, therefore light weight, cheep and reliable

c) multiplying is possible without any problems

d) sensefull applicabel also with few members

e) easy to test and maintain

f) felxibel in case of system changes and extantions.

Due to this qualities we think it's worth to think about it as an alterate for wirering discrete signals in the conventional method or using the databus-system.

Apendix

State of the program, January 1978 :

Using standard MSI components about 16 digital members and some analog members, have been built. (see fig. 1). Then using little hard-wired test boxes (see fig. 2) the system was checked out in all modes. Limited IMC tests will start very soon.

Some of the functional models were delivered to VFW to be tested in a remote power control system using the electrical power RIG of the VFW 614.

Some other functional models are being tested in VDO for use in bigger land vehicles.

In addition a databus terminal MIL-STD 1553 A (modified) will be extended to commect with the discrete bus system.

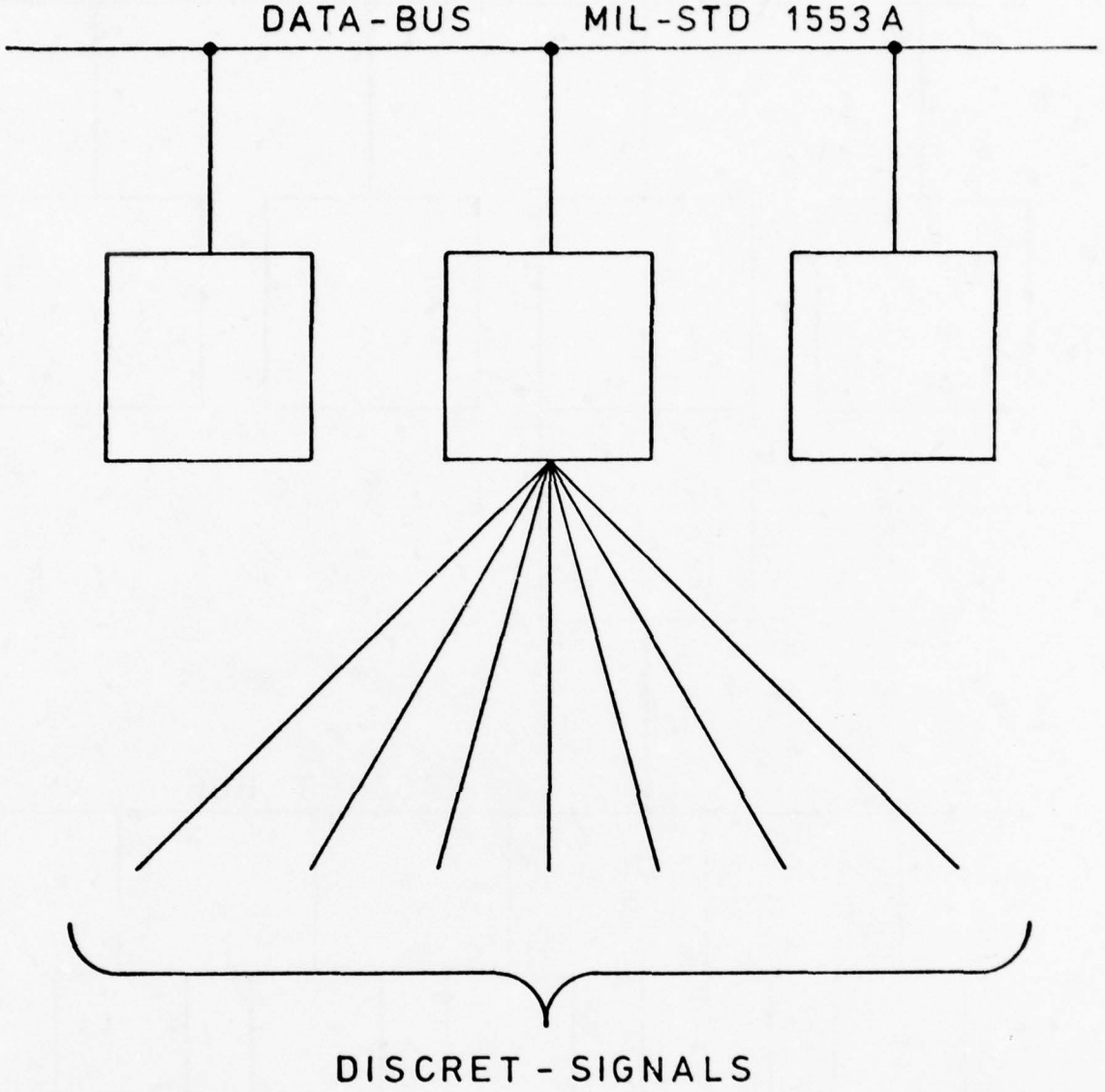



FIG.1

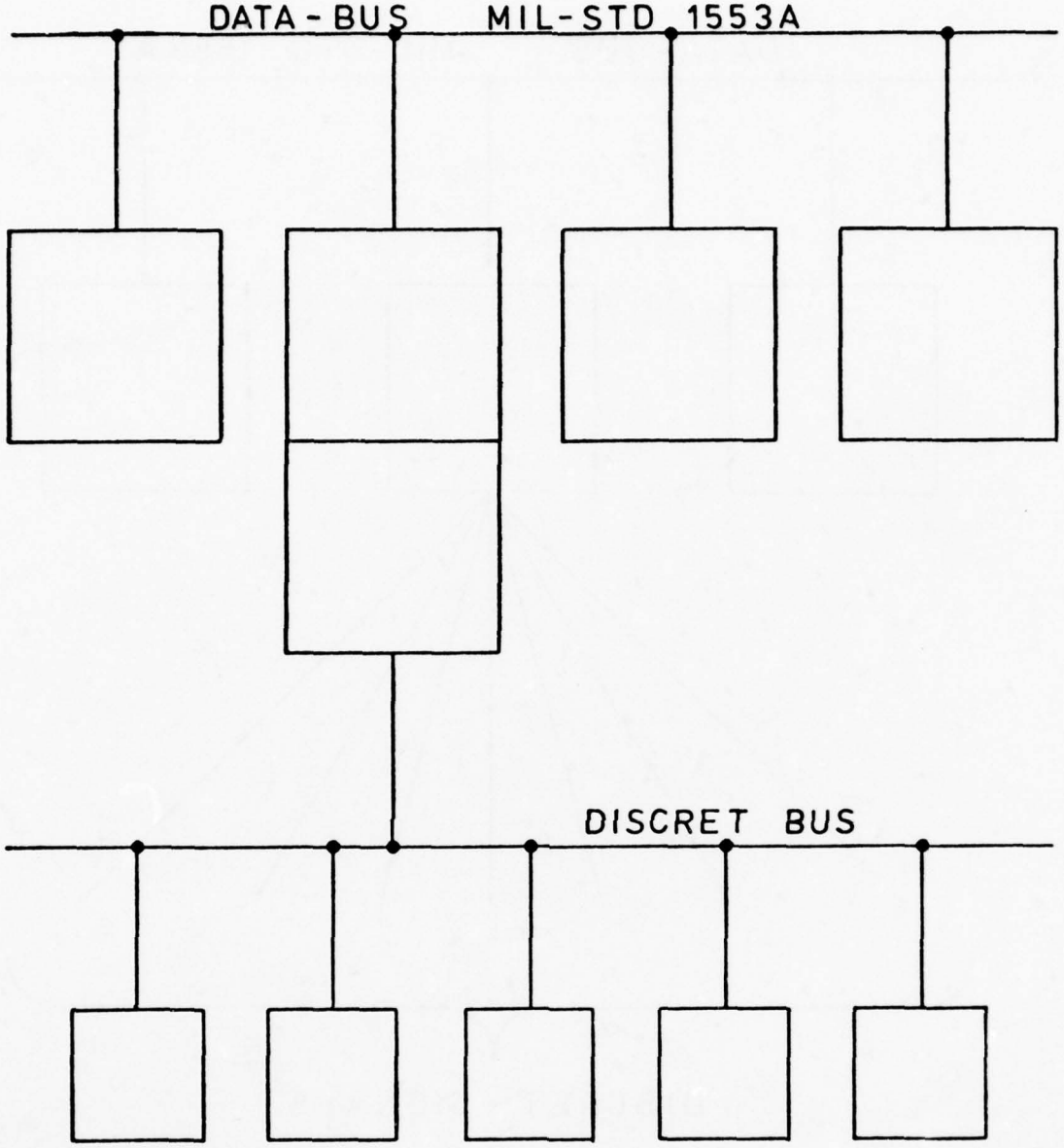
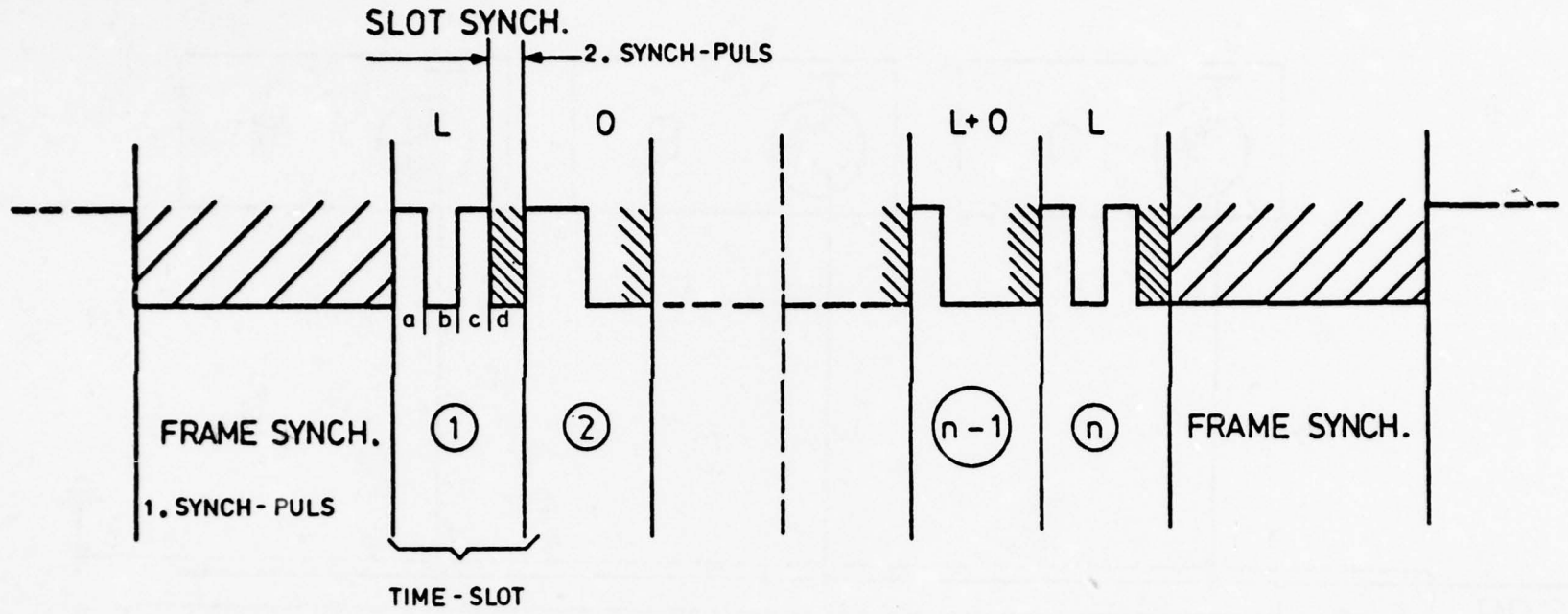



FIG.2




SYSTEM-CLOCK : 10,24 kHz

BIT – CLOCK : 2,56 kHz

FRAME-TIME : 100 ms

FIG. 3 SIGNAL-PULSDIAGRAM ON THE BUS

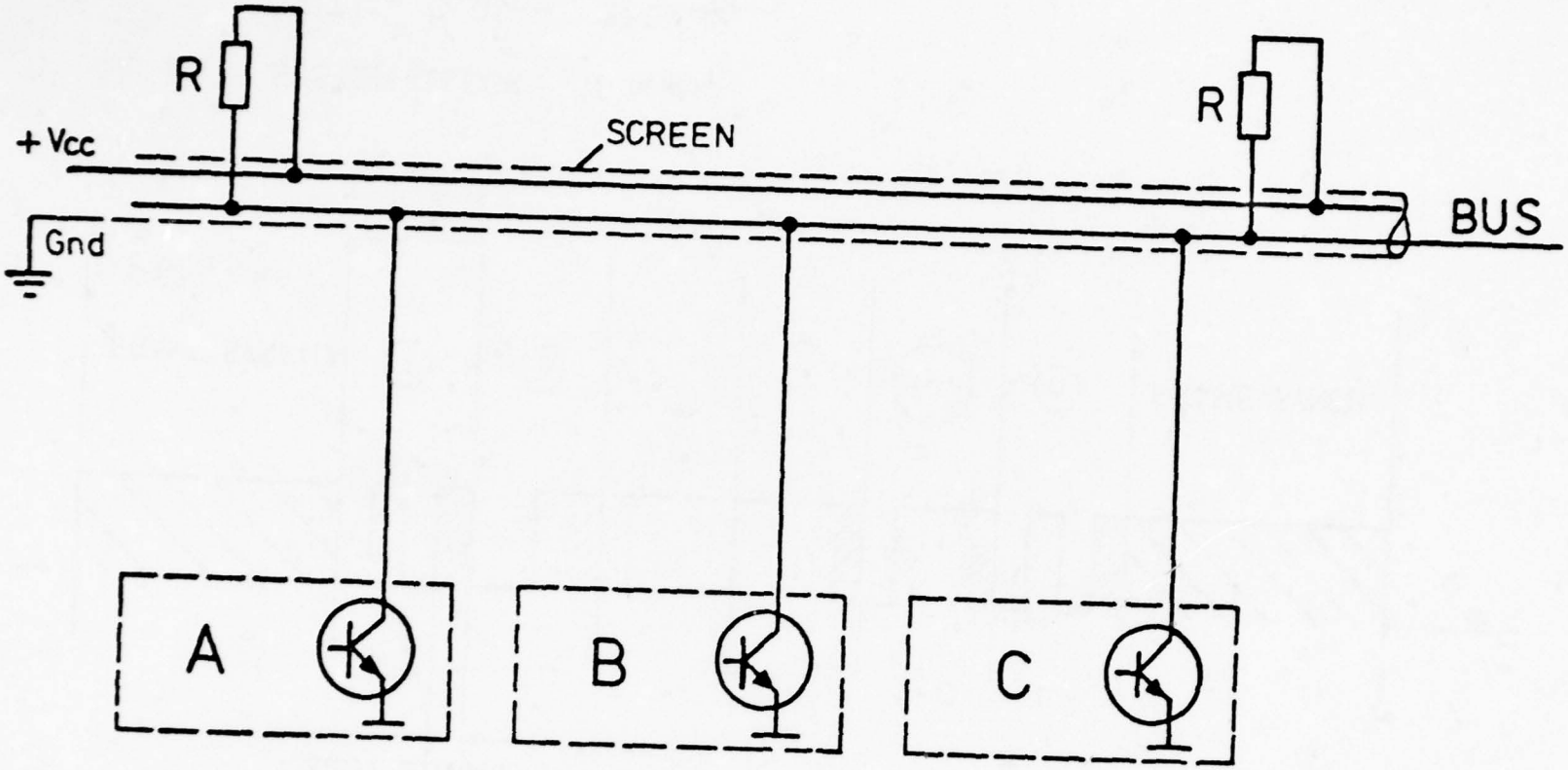



FIG. 4 BUS DRIVER

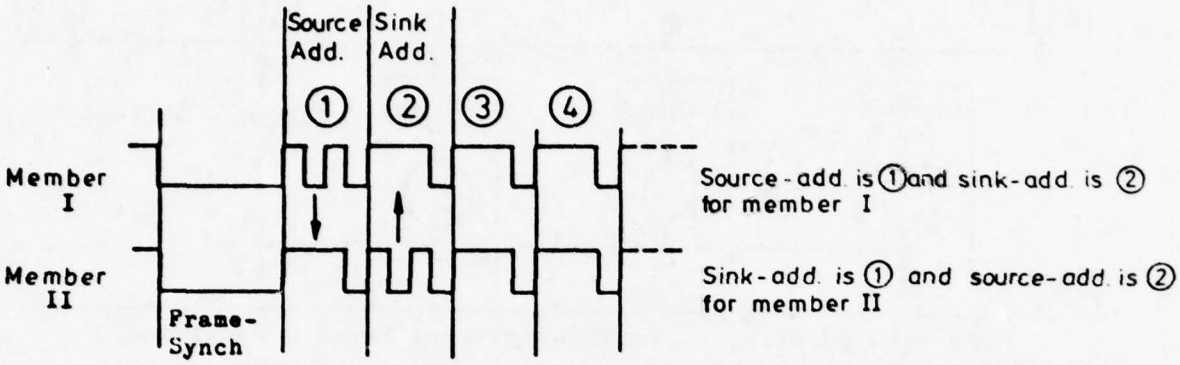



FIG. 5a Correlation of Source-and Sink-Add. (Puls diagram)

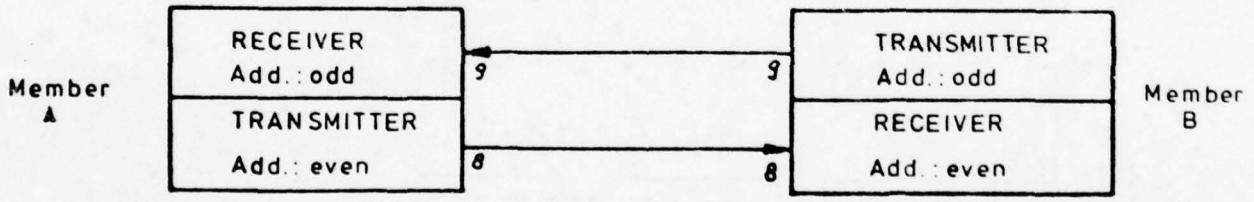



FIG. 5b Correllation off Source-and Sink-Add.

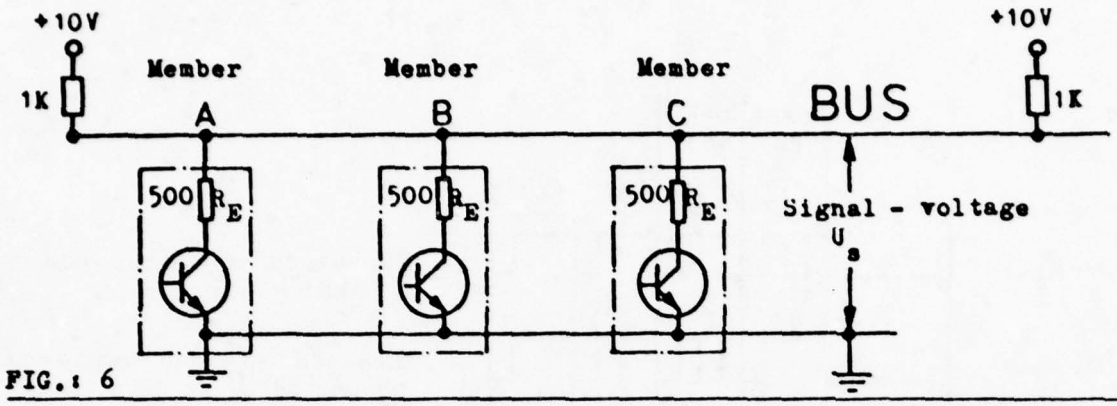
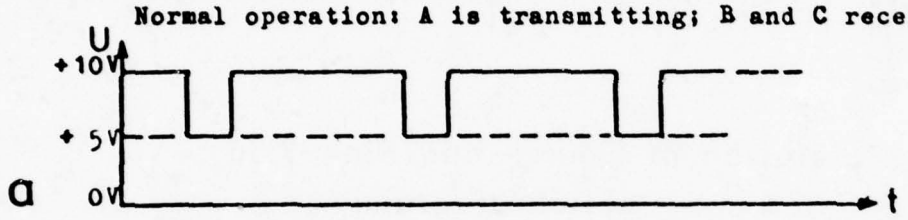
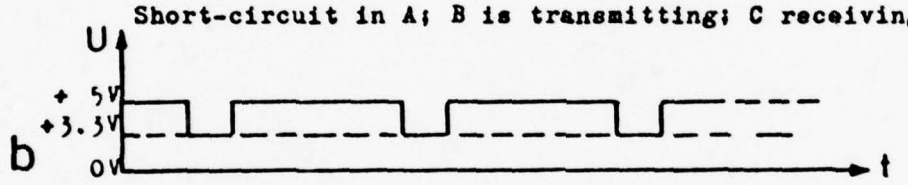



FIG.: 6

Normal operation: A is transmitting; B and C receiving




Short-circuit in A; B is transmitting; C receiving




Short-circuit in A and B; C is receiving

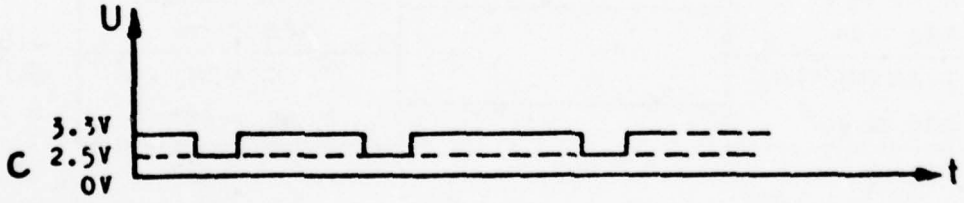



FIG.: 7 Signal on the BUS in normal operation and short-circuit - case

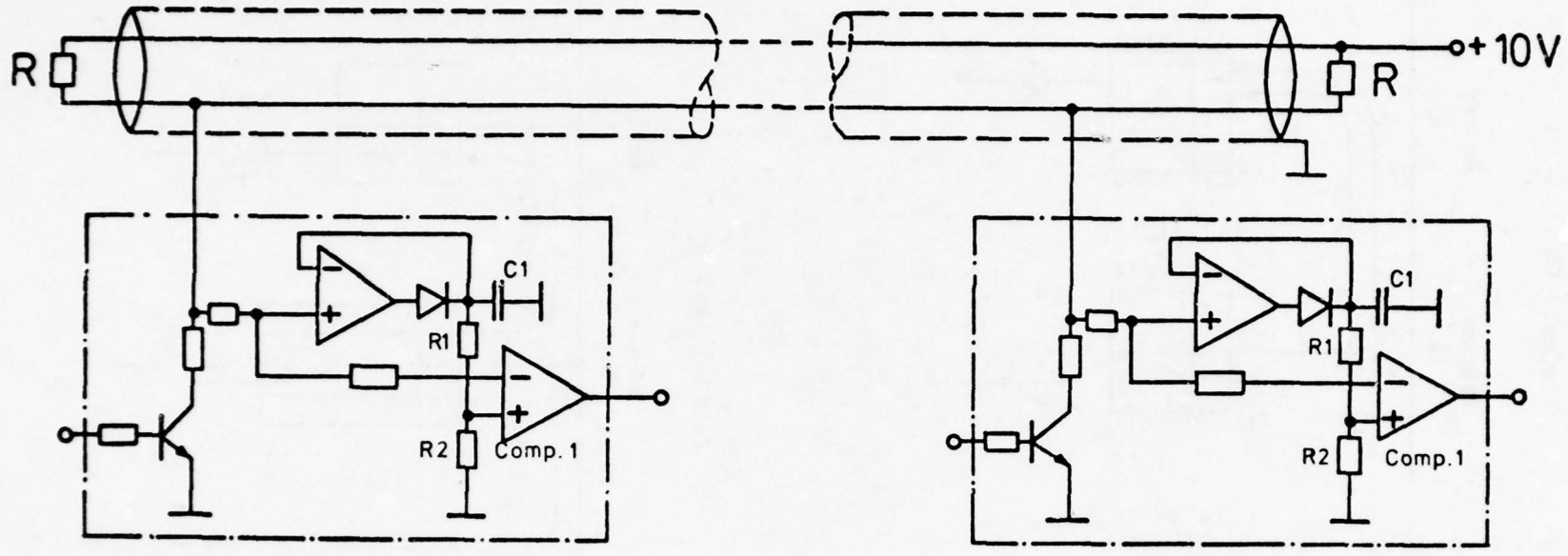



FIG. 8 RECEIVER - TRANSMITTER

logical OR

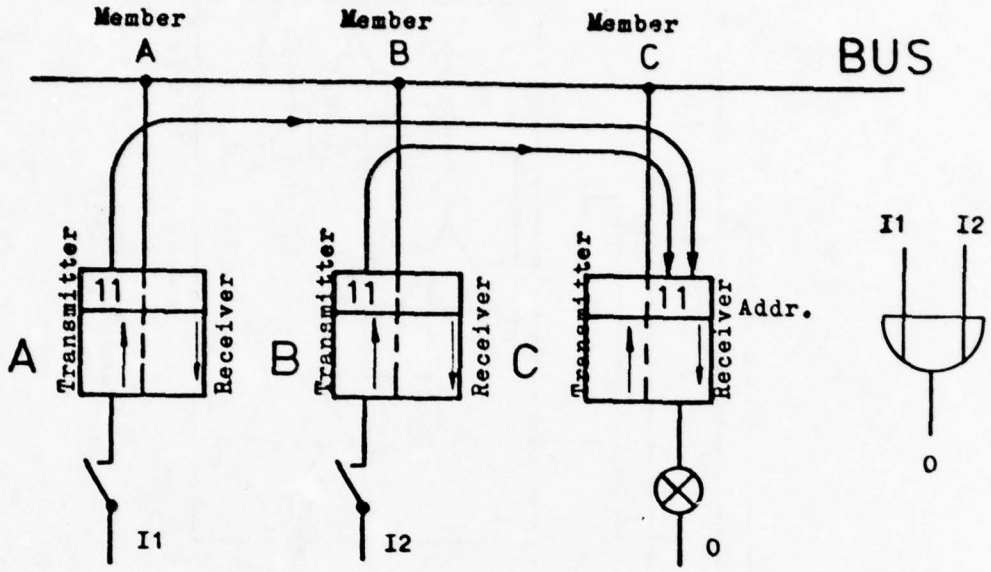



FIG.: 9

logical AND

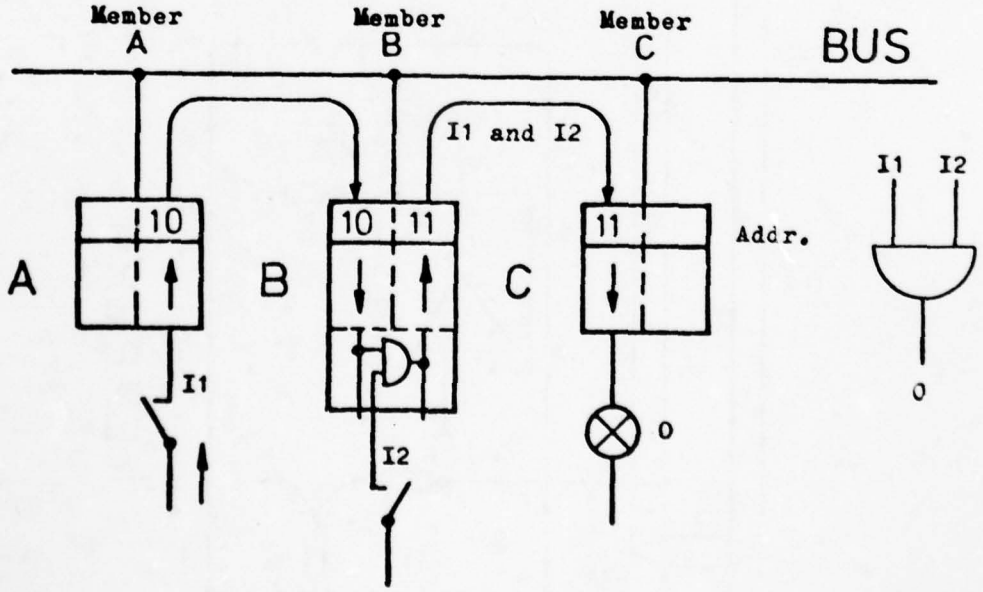



FIG.: 10

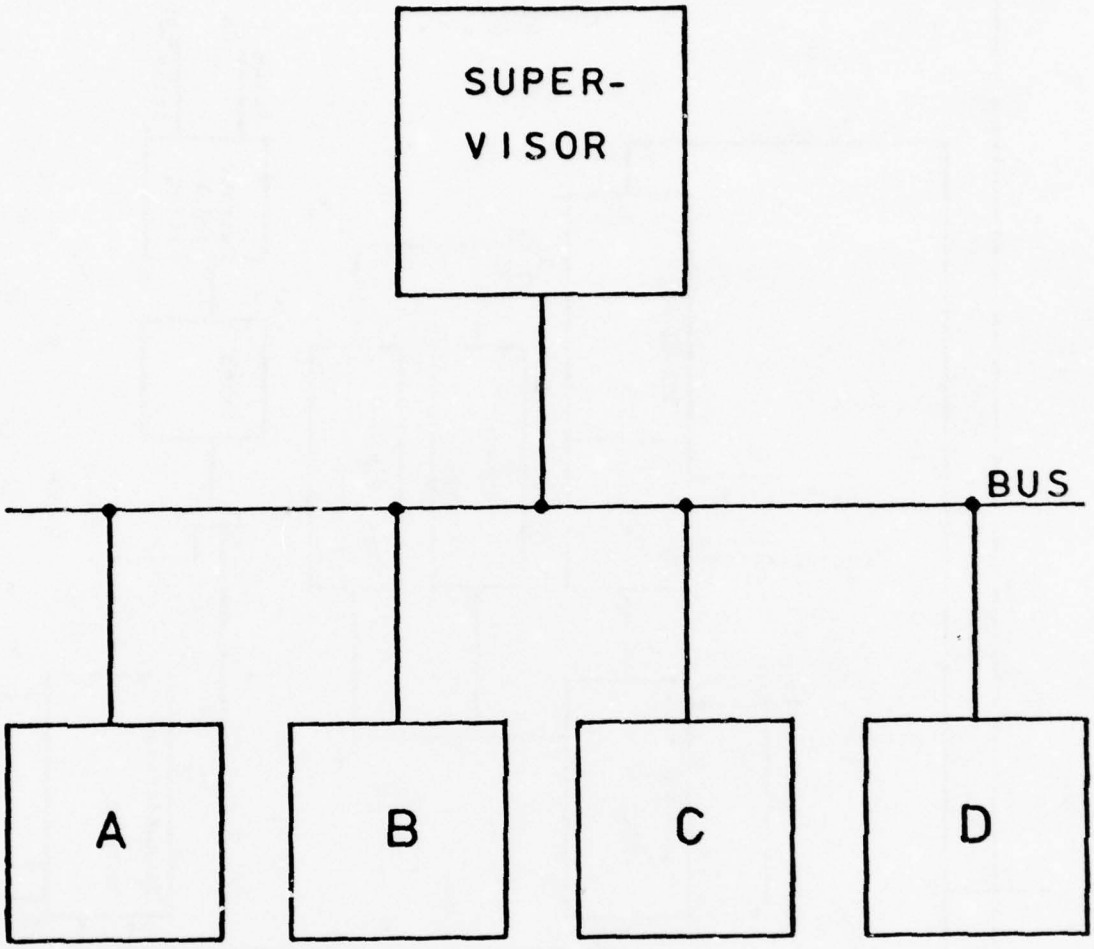



Abb. 11

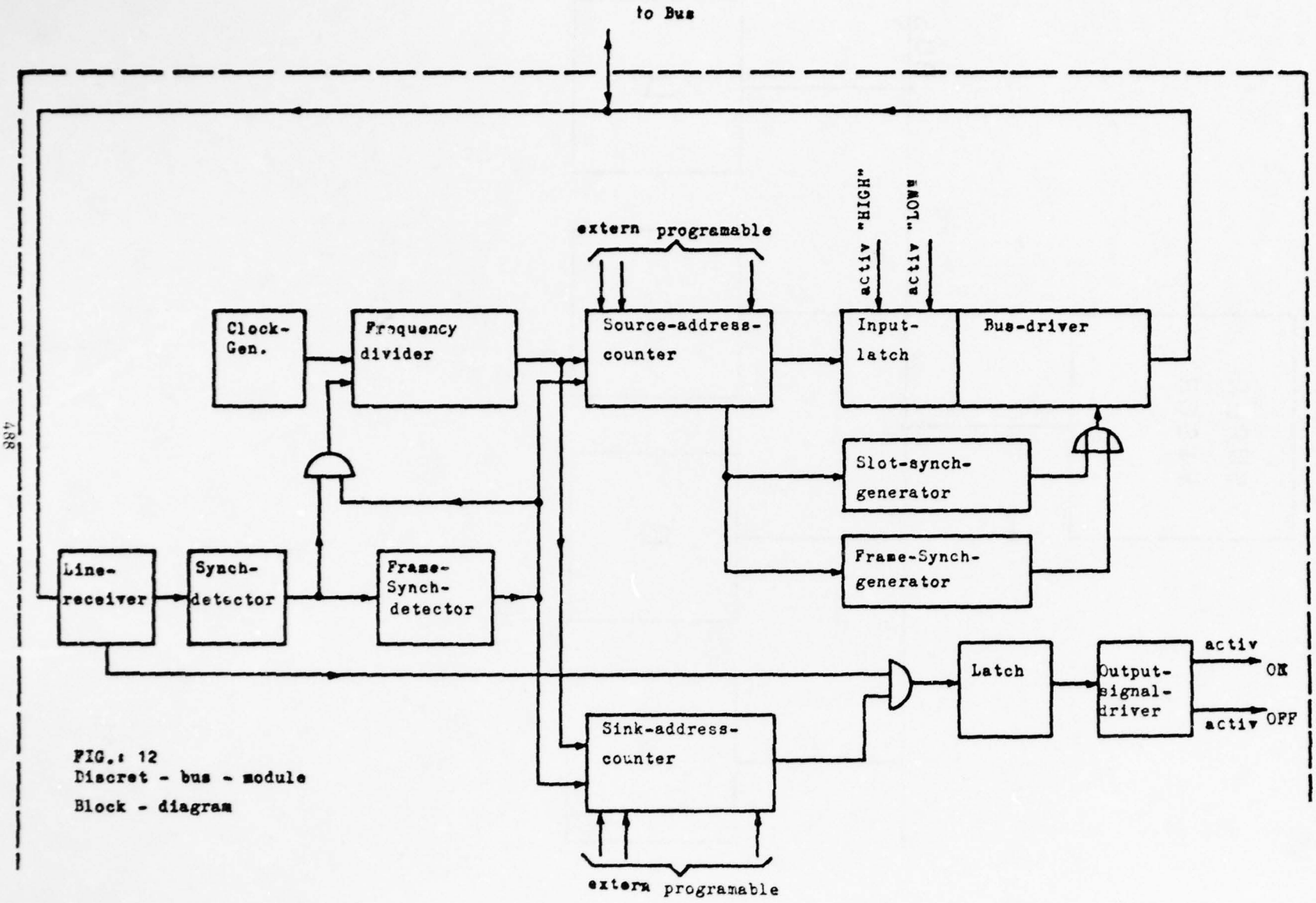



FIG.: 12
Discret - bus - module
Block - diagram

# DEVELOPMENT OF THE MIL-STD-1553 MULTIPLEX APPLICATIONS HANDBOOK

C. Ray Turner
The Boeing Company
Seattle, Washington 98124

## ABSTRACT

This paper is intended to give the reader some concepts and expectations of the content of the Air Force Systems Command sponsored Multiplex Applications Handbook.

## BACKGROUND

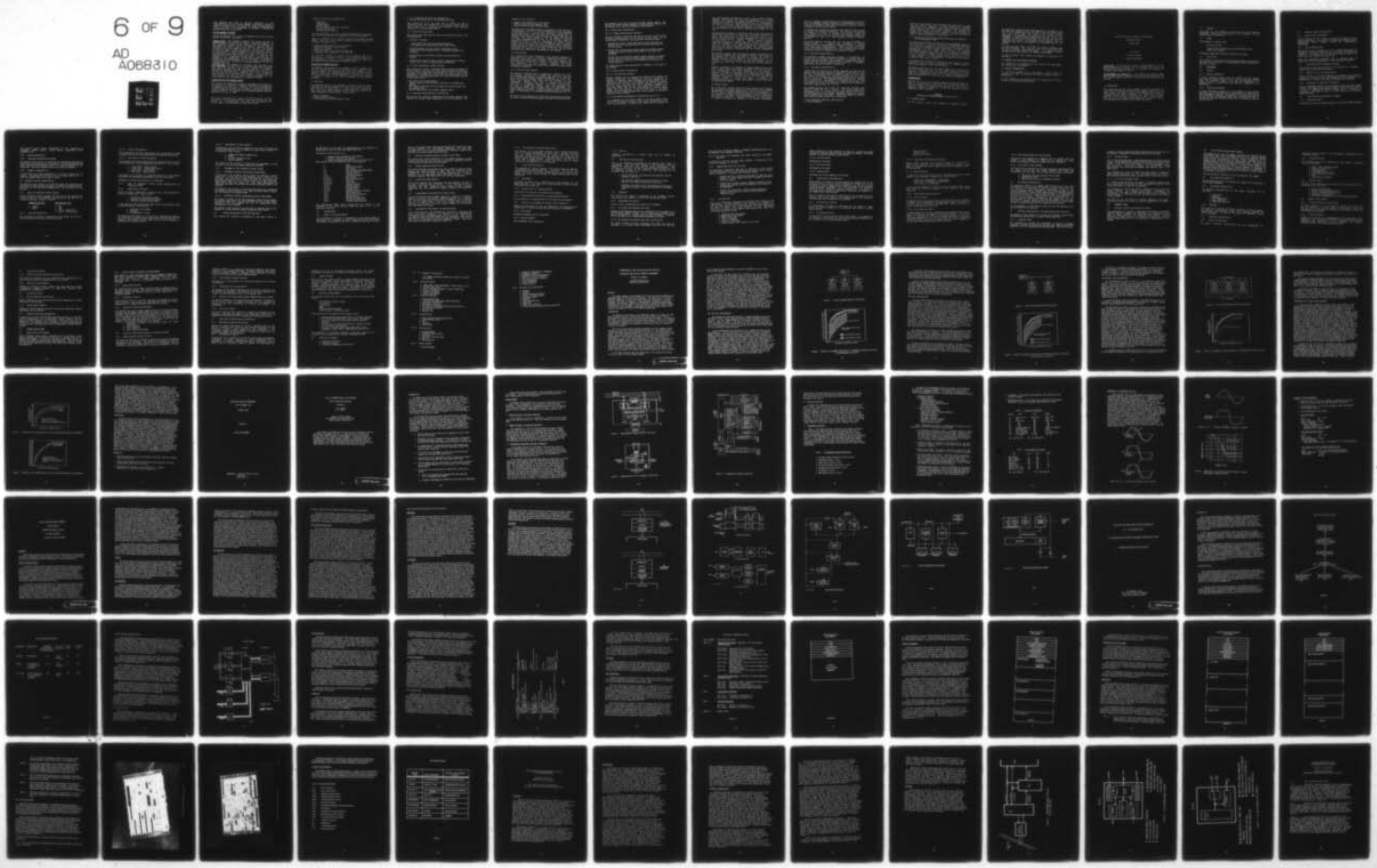
The draft MIL-STD-1553B (16 June 1978) states "Even with the use of this standard, subtle differences will still exist between multiplex buses used on different aircraft due to the options allowed in the standard". (Underlining by the author of this paper.) Also the draft states, "These designer selected options must exist, so as to allow the necessary flexibility to assemble a custom multiplex system from the functionally standard parts and to program the standard electronic functions in order to provide a control mechanism, traffic patterns, redundancy, and a viable degradation concept".

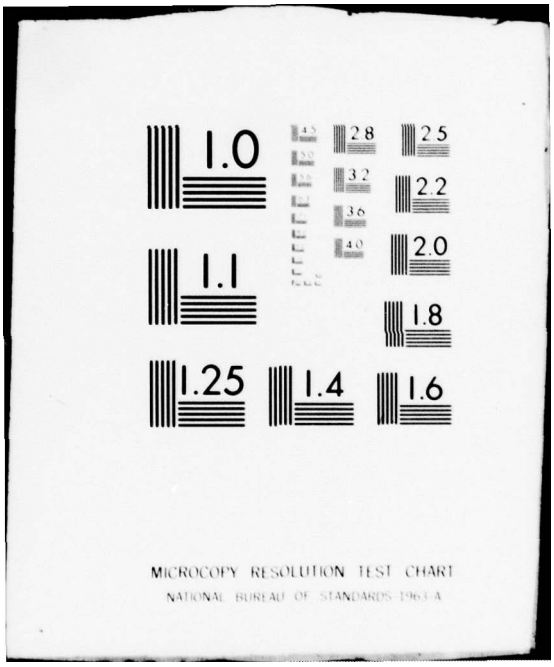
The question is: how to achieve standardization without overcontrol, excessive cost, or stifling of technology. MIL-STD-1553 was written to help solve a problem: proliferation of digital signal interfaces of various aircraft avionics systems. While the standard does reflect very significant industry input and agreement, the actual statements are necessarily brief, and independent of any particular multiplex system architecture, hardware, or software. And it is permissive. Standard digital signal interfaces are being used, and the integration approach of 1553 is well accepted, but minor application design deviations which can cause interoperability problems are occuring.

In view of the fact that: (a) the rationale for the standard is not in the standard; (b) past experience and "lessons learned" in implementing MIL-STD-1553 is not generally available; (c)standards

AD-A068 310 AERONAUTICAL SYSTEMS DIV WRIGHT-PATTERSON AFB OH F/G 17/2
PROCEEDINGS OF THE AFSC MULTIPLEX DATA BUS CONFERENCE (2ND), HE--E
OCT 78 E C GANGL
UNCLASSIFIED ASD-TR-78-34-VOL-2 NL

6 OF 9
AD A068310




1.0
1.1
1.25
1.4
1.6
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.8
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

with handbooks have been an effective combination in other technologies; the Air Force Systems Command is sponsoring a program to prepare a handbook which will present the design considerations and trades which need to be accomplished to implement a MIL-STD-1553 multiplex system for aircraft.***

## WHY THE HANDBOOK IS NEEDED

Several anticipations exist of what the handbook will do for industry and for the Air Force, as sponsor.

Industry View: The aerospace industry that serves the DOD is organizationally quite dynamic and flexible. Reassigned managers, who have appropriate digital systems experience but no background in multiplexing for aircraft will use the handbook. Engineers also get reassigned. And college graduates, too, will need help. The handbook will help get them up to date and productive. In each case, tutorial sections are needed. Existing designers of 1553 systems will want to see information on systems designed by others; how was the data processing partitioned, what sensors were used, etc. Therefore, the handbook will be a technology transfer medium. Engineers who are designing a new implementation of 1553-type system or designing a retrofit will find the handbook useful as a reference. Checklists and "considerations" will be useful in planning management as well as technical activities.

Air Force View: The handbook will forcefully cover the concept that more interoperability could have been achieved among previous systems than what was achieved. The economic benefits of standards, specifically multiplex interfaces and integration per 1553 will be extolled with missionary zeal. Imagined, but not real, difficulties of using 1553 will be anticipated and addressed. Air Force SPOs and other agencies that fund and control system development will see clearly the need for and advantages of integration via 1553.

## ADMINISTRATION OF THE HANDBOOK EFFORT

The Air Force has authorized a period of approximately six months for the contractor to complete the handbook, including the incorporation of comments from the Air Force. Following this, the SAE A2K Committee on Multiplexing For Aircraft will be asked to sponsor and perform a review by industry. An acceptable and coordinated handbook should be available by the fall of 1979.

---

*** Contract F33615-78-C-0112 awarded to The Boeing Company; with SCI Systems Inc., as subcontractor. This paper was not prepared with contract funds. Views expressed are those of the author.

The major sections of the handbook are:

Background
System Design
Hardware Design
Error Prevention, Detection, Correction
Software Architecture
Multiplex System Examples

The contractors have selected "principal investigators" for each major section of the book who will be responsible to the program manager.

Under the leadership of the Multiplex Applications handbook program manager, who also ensures overall quality of the sections, the principal investigators will:

- o Prepare annotated outlines of each section;
- o Supervise, and prepare draft sections;
- o Review all sections;
- o Attend and support the technical reviews; and
- o Adjudicate review comments for their sections.

The Air Force supplied the outline for the handbook, and it was referenced in the technical tasks to be performed by the contractor. See Appendices A and B to this paper (Technical Tasks, and MIL-STD-1553 Multiplex Applications Handbook Outline)

## DISCUSSION OF HANDBOOK SECTIONS

The handbook, when prepared, will probably be 300 or so pages. The purpose of this discussion is to give readers of this paper a "feel" for the emphases and expansions that have been planned for the handbook sections. The following expansions of portions of the outline carry the same outline titles as Appendix B to this paper. All of this is subject to change as the actual development of the handbook progresses.

### II. Background

The background section of this handbook is intended to give the casual reader and the high level manager a feeling for the historical factors which have led to multiplexing as a viable approach to integration. This section will address some of the key issues that lead to the system integration via multiplexing. These include:

- o Need to multiplex;
- o Trends in avionic sensors;
- o To what extent should multiplexing be used;

- o Is it compatible with other multiplexing; and
- o How were the conclusions in the standard determined.

This section will be at the right level of detail, so that an interested reader will proceed into the detailed chapters and the casual reader will be satisfied that multiplex is an inevitable solution to his system design problem.

II. A.1 Need for Multiplexing

This section will show that the need for multiplexing occurs in the following manner:

- o The need to integrate
  - large penalty associated with dedicated sensors
  - intercommunication requirements high between sensors
- o System performance (mission success) improvements can be accomplished using integration due to redundancy approaches now available.
- o Standardization is achievable using the prescribed data bus approach.
- o Standardization provides sensor interface compatibility reducing initial cost as well as allowing system retrofitting.

II. A.2 Move from Analog to Digital Circuits

This section will provide a technology update on what has happened to avionic sensors in the past five years. The discussions will show that the advent of microprocessors for computation as well as logic device replacement has brought the sensors from the analog world to the digital world. The technology survey will provide the reader with the following conclusions:

- o More sensors are becoming digital;
- o Sensors, because of microprocessor technology, are becoming more capable;
- o Most sensors information can be provided via a serial digital data bus standard; and,
- o Sensors using the data bus standard integrate easily.

II. B. Rationale Behind Features of MIL-STD-1553

This section will include an explanation of each major section of the standard and rationale (where appropriate) for the requirements. The

purpose for this section is:

- o Expand on the explanation in the standard
- o Explain reasons for the approach; and,
- o Show differences between 1553A and 1553B

Currently we are discussing several alternatives to this section of the Background. One view is that the explanation of the very meaty, concise standard be in a section after discussion of System, Hardware, and Software Design. This view holds that those sections will (or should) contain tutorials which will help the reader understand the rationale. The counter argument is that the handbook is concerned with design implementation of 1553, so the rationale should be presented initially so that the requirements are firmly in mind. Another view is that a tutorial on an information transfer system be presented with its physical and functional architectures, transmission media, and protocol. This will provide information so that the reader understands the reasons for the choice of the bus, the electrical characteristics, and protocol requirements in the standard.

III. System Design

This section will provide information to the system's integrator to help establish the level of multiplex integration required for the particular application being analyzed. The basic premise that multiplexing for integration is a viable cost effective technique has been demonstrated for many vehicles. Therefore, the system integrator's real problem is to establish by analysis the proper level of multiplex integration.

III. B. Life Cycle Cost Analysis

The decision to multiplex must be supported by objective life cycle cost analyses. Generally it is conceded that acquisition costs may not be significantly reduced by data bus integration. It is also well known that military airplanes produced decades ago are still in use, having been modified many times for new missions and new requirements. Therefore, the level of change activity after acquisition is an important parameter. In the case of multiplexing, the ease of modification usually shows LCC advantages over point-to-point systems, even for modest levels of change activity and limited numbers of airplanes. Production quantity may be an important factor in determining if a nonstandard multiplexing approach is to be used.

This section of the handbook will identify the factors which influence the choice of multiplex system designs; how these factors affect LCC;

and parametric cost data on current multiplex system elements. The section will advise system designers to become familiar with the LCC disciplines, and to seek the assistance of professionals.

III. C. Architecture Considerations

III. C.2. Computation/Controller Locations

The bus controller's functional type will be discussed in this section including characterization of the methods of bus control and the various types of controllers. The types to be discussed include:

- Systems that utilize a stand-alone dedicated bus controller that performs all of the required functions to operate and maintain a data bus system.

- Systems that utilize general purpose computers performing system integration functions and the I/O function of bus control and management.

- Systems that utilize multiple general purpose computers to operate as bus controllers as well as perform specific system functions. The bus control in this case can be dynamically allocated to any capable processor.

The understanding of these applications is essential to the reader of the handbook.

III. D. System Control Considerations

III. D.1. Bus Controller Software

Software designers need to understand the system control methods, the hardware architecture and capabilities and the partitioning of functions to processors before messages and message pointers can be constructed. This section of the handbook will present the general method by which table driven bus control software is designed, and will give several specific (but still generalized) message lists, timing and control data and the corresponding bus control software design. With this material, it will be quite clear to the system designer and the software designer what their interface is, and how to design bus control software.

III. E. Bus Network Considerations (proposed addition to III)

It is suggested that this topic be added to the system design outline because of the substantial impact of bus network configurations on bus error rate performance. In other words, error rate is a function of

the system network configuration as well as bus interface hardware designs. Unfortunately, bus error rate performance is not easily predicted for a specific bus network because of the many variables affecting the quality of a waveform from the time it is transmitted to the time it is received. These variables can be broadly categorized as: (1) factors affecting waveform integrity at any given stub and (2) characteristics of the receiver affecting data reception at that stub.

MIL-STD-1553 has attempted to specify the more obvious characteristics which affect error rate but difficulty arises when one tries to specify such characteristics as the type of receiver filter or sync detect algorithm required. Dictating implementation of design cannot be the intent of the standard. Instead, receiver interface requirements rather than design have been devised to "guarantee" proper bus performance. New requirements are still being added to this list, the most recent being a minimum shunt inductance value imposed on the receiver coupling transformer, and a minimum waveform zero crossing distortion requirement that the receiver synchronizer must accept.

MIL-STD-1553A requires that the multiplex system successfully work with up to 33 terminals and stub lengths to 200 feet. This section of the handbook will contain guidelines to indicate the effects on waveform distortion as the result of bus configurations. It is difficult to accurately predict the distortion present and the resulting bit error rates to be expected with existing empirical data.

Computer simulations are available to develop parametric data that will be used to define the waveform distortion to be expected from a given configuration. Data from simulations will be reduced to envelopes of permissible cumulative stub lengths and trunk lengths. Separate cases will be developed for bus configurations employing transformer stub couplers and those that do not. The data will be primarily intended to inform the reader of preferred bus configurations and give him some understanding of system sensitivity to stub and trunk lengths.

IV. Hardware Design

One of the major problems facing the system engineer when designing and specifying a multiplex system is the transition from the standard (MIL-STD-1553) to equipment specifications. The normal approach is to write a multiplex system specification based on an interpretation of the standard in the areas of system operational protocol, architecture and electrical interfaces. Invariably differences in interpretation or opinions concerning the adequacy of the standard requirements

result in equipment interoperability and interchangeability problems. Also, the equipment engineer attempting to design hardware for use on a variety of programs is faced with the problem of different interface requirements for the various systems.

The handbook will provide the guidelines and trade-offs which will aid in reducing the inadvertent deviations. This section of the handbook will address the areas of hardware specification, design and implementation so that the more cost effective "standard" hardware can be developed and employed in future programs utilizing a multiplex data bus.

The outline, Appendix A (Section IV) covers all items of hardware in a typical system. It is suggested that subparagraph B, entitled "Stubbing and other considerations in laying out the bus", be modified to be "Bus Coupler Characteristics". This discussion will center around the electrical characteristics of the bus network and coupler, including the effect of stubbing on the transmitted signals.

IV. A. Transceiver Characteristics

The transceiver includes three major elements: (1) transmitter, (2) receiver and (3) coupling network to provide an interface with the twisted shielded pair cable. The handbook will contain Design Manual, Design Tips and similar material usually available to designers, specifically tailored to design of transceivers.

IV. B. Bus Coupler Characteristics

A great deal of confusion has existed regarding the need for and characteristics of a bus coupler for connection of the terminal to the main bus. MIL-STD-1553A requires a separate coupler box to be used. Two couplers are defined: (a) isolation resistors only for a short stub (1 ft.) and (b) transformer and isolation resistors for long stub (20 ft.) connections. The proposed MIL-STD-1553B requires a coupler with resistors and a transformer if a coupling box is used. The transformer turns ratio is not defined in 1553A and 1553B specifies a turns ratio range of from 1.41:1 to 1.5:1.

The handbook material will relate the trade-offs associated with length of stubs and use of couplers. The network configuration considerations will be included in the System Design Section (III). The characteristics of the coupler transformer required to maintain signal integrity will be described. Coupler box design and connector type selection for EMI control will be considered.

V. Error Prevention, Detection, and Correction
V. A. Error Detection

Sources of errors within a given multiplex bus system can be divided into two separate categories: a) analog related errors primarily originating in the tranmission media, and b) digital or functional errors resulting from faults within terminals. The handbook will discuss these sources in a manner intended to alert the reader to various error situations that are likely to occur and describe the impact on systerm performance of these errors.

V. B. Methods for Calculating and Measuring Detected and Undetected Bit and Message Error Rate

This section will define the terms bit, word, and message error rate and relate them to a multiplex bus system. MIL-STD-1553A requires a receiver and one transmitter to operate in the presence of burst noise in a relatively uncomplicated bus configuration. Problems of test repeatability and test length have resulted. The proposed MIL-STD-1553B modifies the error rate test to exclude transmission line effects and has redefined noise to be white Gaussian. This type of test is viewed as a "figure of merit" test of the receiver, not of a system. However, it has the advantage of repeatability and requires a relatively short time to complete.

The handbook will describe the evolution of and rationale for noise testing and current attempts to standardize on a common concept of error rate as applied to and isolated receiver.

Determining error rates of a bus system rather than a receiver is a far more encompassing task. The bus network section will include reference to noisy buses and give statistical indications of bus error rates for a hypothetical system, but it should be emphasized that predicted system word error rates are only as good as the model used.

ACKNOWLEDGMENT

Material from the Air Force Systems Command RFP F33615-78-R-0112 was used in the preparation of this paper; and also extracts from the Technical Proposal jointly prepared by Boeing and SCI. The author is especially indebted to members of the SAE AZK committee in those companies for their assistance.

APPENDIX A
MULTIPLEX APPLICATIONS HANDBOOK F33615-78-C-0112

4.0 TECHNICAL TASKS

4.1 The contractor shall review Appendix A (Handbook Outline),

MIL-STD-1553A, proposed MIL-STD-1553B (Appendix B) and submit to the Air Force his proposed outline two weeks after the contract start date. The outline shall list issues to be covered, indicate sub-issues to be developed, and predicted levels of detail to be expounded. A technical review meeting shall be held at the prime contractor's facility three weeks after the contract start date to coordinate the outline. The multiplex system examples that the contractor shall use in the handbook will be agreed upon at this meeting.

4.2 The contractor shall review DOD and industry programs using MIL-STD-1553 systems. The contractor shall also review applicable studies on multiplex systems and hardware. As a minimum, programs reviewed shall include the F-16, B-1, DAIS, F-18 and LAMPS.

4.3 The contractor shall produce the handbook in accordance with Appendix A.

5.0 REPORTS, DATA, AND OTHER DELIVERABLES

All reports are in accordance with the Contract Data Requirements List, DD Form 1423, Attachment #1.

6.0 SPECIAL CONSIDERATIONS

6.1 Air Force personnel will be available to discuss with the contractor the areas of philosophy and rationale behind features of MIL-STD-1553.

6.2 The contents of the handbook shall be nonproprietary and shall become Air Force property upon delivery.

# MIL-STD-1553 Multiplex Applications Handbook

Annotated Outline

October 1978

C. R. Turner

The Boeing Company

Seattle, Washington

Introduction: This outline was prepared as CDRL Sequence No. 1 by The Boeing Company and SCI Systems, Inc. It represents the status of the Multiplex Applications Handbook, and is included in this volume for information, and comment if desired.

Acknowledgement of Sponsorship: This outline was prepared under Contract F33615-78-R-0112 sponsored by the United States Air Force, Air Force Systems Command, Aeronautical Systems Division, Wright-Patterson AFB, Ohio

## 1.0 INTRODUCTION

The introductory material in the handbook is required to acquaint the reader with the scope and purpose of the handbook, authority for the book, status, and general organizational approach which cannot be inferred from the Table of Contents. This section will be short, pertinent, and will serve to encourage the reader to read the whole book, or to find the section which relates to his specific concern. The introduction will contain the following subsections:

## 1.1 Purpose

The purpose of this handbook is to present all of the necessary guidelines, trade offs, considerations, and procedures for the effective use of MIL-STD-1553.

## 1.2 Scope

This handbook is concerned with:

Military aircraft,

Digital serial multiplexing using MIL-STD-1553 as the controlling standard.

Differences between MIL-STD-1553A and 1553B are included.

## 1.3 Definitions

This section will help the totally uninitiated reader by presenting a few terms that are essential to understanding multiplex technology as applied to aircraft.

# 2.0 BACKGROUND

## 2.1 History

### 2.1.1 Vehicle Integration

A brief historical overview will be provided in this section describing the vehicles, their missions, and the integration approach used. This section's purpose is to show that integration is a necessary factor in aircraft and serial data bus integration is the method used.

### 2.1.2 Standards Development

A brief historical overview will be provided in this section describing the military approach to controlling this integration by the development of standards. The purpose of this section is to show that the standard interface philosophy which yielded MIL-STD-1553 is not new but has been developing along with the integration approach and application.

2.2 Technical Trends to Multiplexing

2.2.1 Vehicle Mission Performance

A brief overview will be provided to indicate the complex missions being required today. The purpose is to indicate the extensive use of sophisticated sensors, resource sharing and interconnection requirements.

2.2.2 Vehicle Modifications

The purpose of this paragraph will be to indicate the flexibility inherent in multiplex system and the continued requirement for flexibility in the future for the two types of vehicle modification which occur, viz: updating and retrofitting.

These two modification approaches will be examined where the modification involves systems with or without multiplex integration.

2.3 Issues Resolved by Developing a Standard

A discussion of the unique position of the MIL-STD as a standard versus a specification will be discussed.

2.2.3 Reduction of Cost by Standardization

A brief overview of the DOD's approach to procurement in the area of standards as it applies to aircraft integration will be presented. The purpose here is to identify the reality and necessity of standards.

2.2.4 Technology as it Applies to Integration

A brief technology overview will be provided to show the trends toward digitally intelligent devices and the potential this provides to achieve a level of compatibility between sensors which is required in integration. The purpose of this is to point the way to the standard selection process.

2.3 Issues Resolved by Developing a Standard

2.3.1 Application Areas

Discussed will be how the standard applies to the real time data transfer

and aircraft system control requirements of such subsystems as navigation, weapon delivery, flight control, propulsion, stores management, etc.

2.3.2 System Architectures

2.3.3 Subsystems Using Multiplex System

The purpose of this section is to show that the standard was developed to satisfy the application and was not arbitrary. A brief discussion of the signal types and characteristics found in aircraft integration will be discussed in relation to estabalishing a serial data bus standard.

2.3.4 Degrees of Compatibility

A brief discussion on what compatibility is and how it applies to the standard. The purpose of this section is to indicate some of the key areas the standard identified to make it useful.

2.4 Information Transfer System Tutorial

The purpose of this section is to show the reader the reasons for the choices of the 1553 bus (from the types of Information Transfer Systems), the choice of the electrical characteristics, and the choice of the protocol.

2.4.1 Types of Information Transfer Systems

Various Information Transfer Systems which are used in the communication of digital data will be identified, and the MIL-STD data bus will be related to the group. These will include:

| COMMUNICATION TYPES | CONFIGURATION TYPES |
|---|---|
| o simplex | o star |
| o duplex | o loop |
| | o serial - single user |
| | o serial - multiple users |

2.4.1.1 Physical Architectures

This paragraph will contain a description of the effects the aircraft physical characteristics had on the MIL-STD design.

2.4.1.2 Control Architectures

This paragraph will contain a description of the various control schemes allowed by the MIL-STD, centralized control and decentralized control.

2.4.1.3 Description of 1553 Architectures

This paragraph will contain descriptions and discussions of the various architectures which can be developed using the MIL-STD. These include:

- o single level - central control
- o single level - decentralized control
- o multi-level - local control
- o multi-level - central control

The purpose of this section is to show the flexibility of the Standard and discuss the advantages/disadvantages of each bus architecture.

Advantages of the bus architecture will be explained.

2.4.2 Types of Information Transfer System Communications and Transmission Media

2.4.2.1 Communication Techniques

Various multiplex communication techniques have been developed for serial bus transmissions. These include

- o Time Division Multiplexing (TDM)
- o Frequency Divison Multiplexing (FDM)
- o Space Division Multiplexing (SDM)

In the same way, data encoding techniques differ from multiplex system to multiplex system. These include:

- o Non-return to zero (NRZ)
- o Bi-Phase
- o Bi-Phase-Return to Zero

The purpose of this section is to show that the communication technique for the Standard considered several types before the TDM, Manchester II bi-phase L was selected, and the reason why the selected technique is appropriate for military aircraft.

2.4.2.2 Requirements for Data Integrity

A discussion of the aircraft environment and the reason for selection of certain characteristics in the Standard to improve data integrity will be presented. These include:

- o Command and response communication
- o Parity
- o Lossless transmission line
- o Passive stubbing

The purpose of this section is to show that the requirements for data integrity in the users environment impacted the Standard.

2.4.3 Management of the Information Transfer System

2.4.3.1 Information Transfer System Control Requirements

A discussion of the mechanism used to control an information transfer system with a serial data bus (no dedicated control lines) will be presented. This section will identify the need for, and methods that have been developed to control functions such as initiate, shut-down, change modes, self test, reset, etc. Once these requirements are identified, the method the Standard has chosen to implement them will be discussed.

The purpose of this section is to show that the Standard has implemented the necessary control functions in a simple efficient fashion, causing few restrictions to the system designer

2.4.3.2 Information Transfer System Message Control Requirements

This section is similar to the above paragraph, because the requirements for message identification will be established and the various methods (message ID, word ID, parameter ID, etc.) of accomplishing these requirements will be presented.

The purpose of this section is to show that the approach taken by the Standard is a simple efficient method, causing few restrictions.

2.5 Review and Rationale of MIL-STD-1553A and B

This section will include an explanation of each major section of

MIL-STD-1553B (in the order of paragraphing of the Standard) and appropriate rationale for the requirements identified.

The purpose of this section is to:

- o Expand on the explanation in the Standard
- o Explain reasons for the approach
- o Identify differences between the two versions (A & B)

Major sections of MIL-STD-1553B that will be discussed are:

| | |
|---|---|
| 1. | Scope |
| 2. | Referenced Documents |
| 3. | Definitions |
| 4.1 | Test and Operating Requirements |
| 4.2 | Data Bus Operation |
| 4.3 | Characteristics |
| 4.3.3 | Transmission Method |
| 4.3.3.5 | Word Formats |
| 4.3.3.5.1.7 | Optional Mode Control |
| 4.3.3.5.2 | Data Word |
| 4.3.3.5.3 | Status Word |
| 4.3.3.5.4 | Status Word Reset |
| 4.3.3.6 | Message Formats |
| 4.3.3.7 | Intermessage Gap |
| 4.3.3.8 | Response Time |
| 4.3.3.9 | Minimum No-Response Time-Out |
| 4.4.1 | Common Operation |
| 4.4.2 | Bus Controller Operation |
| 4.4.3 | Remote Terminal |
| 4.4.4 | Bus Monitor Operation |
| 4.5.1 | Data Bus Characteristics |
| 4.5.2 | Terminal Characteristics |

Any subparagraphs within these sections will be included in the discussion of the 1553B sections identified above, if it is not separately identified.

## 3.0 SYSTEM DESIGN

### 3.1 Scope of the System Chapter

This paragraph will provide an introduction to the system chapter by presenting: (a) a discussion of "where" system design ends and hardware and software design begins; and (b) a discussion of the need for other

topics in the system chapter that do not naturally fall within the scope of other chapters, i.e., multiplexing advantages, life cycle cost considerations, as well as the relationship of multiplex system design decisions to a normal system acquisition cycle.

## 3.2 Multiplex Subsystem Selection (To MUX or Not To MUX)

This section will provide information to the systems integrator to help establish the need for and the level of multiplex integration required for the particular application being analyzed.

### 3.2.1 System Integration Simplification

This section describes the procedure and gives examples of how the system engineer goes about defining a proper level of integration using multiplexing. From this point the system integrator will be able to determine for an application where multiplex provides a simpler integration approach than conventional point-to-point interconnects.

### 3.2.2 System Test Simplification

This section describes the integration and test procedures involved in verifying a multiplex data bus system. It will show how the use of simulation and "hot" laboratory bench mockup provides extended integration time, thus reducing the hardware, software, and integration risk.

### 3.2.3 System Updating and Retrofiting Simplification

This section discusses the procedure required to retrofit an integrated system in a vehicle. The data bus integration approach will be compared with the more conventional point-to-point techniques used in previous years for an example system. Since no bus systems have actually been retrofit, only the type of data and typical examples can be presented here rather than actual cases.

### 3.2.4 Documentation Requirements and Improved Methods

This section discusses the documentation requirements associated with system integration procedures. In addition to the documentation requirements, several automated methods will be discussed. These include two computer aided design tools which have been developed in the last few years to aid in the study of multiplex integration systems. These tools have extensive documentation generation capability.

### 3.2.5 Test Equipment Interface Simplification

This section will review ground readiness test, inflight performance testing, and ground maintenance and repair test as applied to the data bus. The methods of monitoring bus traffic, the level of test achievable, and the analysis of the data collected will be presented for several general system examples. Convenient use of the data bus by flight test instrumentation to monitor and collect data on systems involved in the integration will be illustrated.

## 3.3 Life Cycle Cost Analysis

This paragraph will contain examples of LCC factors which are important in determining how much multiplexing is beneficial. This section will identify and discuss the factors influencing choice of multiplex system designs, and advise system designers to become familiar with the LCC disciplines.

### 3.3.1 Cost Models

The models available will be identified with trade parameter sets and output matricies. This section will inform him how to propose system alternatives which reduce LCC.

#### 3.3.1.1 Point-to-Point vs. Standardized MUX

Significant LCC drivers will be identified and compared.

#### 3.3.1.2 Non-Standard Multiplexing vs. Standardized Multiplexers

The cost benefits of standardization will be identified and compared.

### 3.3.2 Specific Cost Breakdown for Typical Aircraft Operation

Examples and explanation of the cost components in the postacquisition time period for a typical aircraft multiplex system will be presented.

#### 3.3.2.1 Maintenance

Maintenance parameters will be identified.

#### 3.3.2.2 Logistic Support

These parameters will be identified.

3.3.2.3 Retrofit

Parameters applicable to a retrofit effort will be examined for suitability.

3.4 Architectural Considerations

This is an introductory paragraph to the material that follows in suparagraphs. The flexibility inherent in the multiplex data bus approach to integration allows multiple system architectures. These architectural questions, and their associated design alternatives, (use of redundancy, location of the controller, load splitting) will be discussed. The key issues are:

- o System architectural considerations and multiplex data bus standard flexibility,
- o Bus controller mechanization and the effects on system operation and system architectural issues; and,
- o Redundancy as applied to the interconnection of data paths - concerning the number of buses and the method of redundancy used.

3.4.1 Redundancy

This section will contain a discussion of the redundancy concepts associated with the elements of the MIL-STD-1553 data bus system.

3.4.1 Nonredundant Bus System

Risks of using a nonredundant bus system will be examined.

3.4.1 Bus System with Redundant Data Paths

Examples of the different types of bus redundancy will be presented as in the difference between the DAIS avionics configuration and DAIS flight control configuration. Examples and associated philosophy and benefits will be included to aid in understanding redundancy of busses.

3.4.1.3 Bus System with Redundant Data Paths and Controllers

The remote terminal interface electronics area where the level of redundancy is a hardware design requirements issue will be discussed

along with the differences between a redundant transmitter/receiver and the completely dual redundant interface.

3.4.1.4 Bus System with Redundant Data Paths, Controllers and Remote Terminals

In addition to the bus and the remote terminal, a discussion of bus controller redundancy is included.

3.4.2 Computational/Controller Locations

The controller's functional type will be discussed in this section including characterization of the methods of bus control and the various types of controllers. The types to be discussed include:

- o Systems that have a stand-alone dedicated bus controller which performs all of the required functions to operate and maintain a data bus system.
- o Systems that use general purpose computers performing system integration functions and the I/O function of bus controller and management.
- o Systems that use multiple, general purpose computers to operate as bus controllers as well as perform specific system functions.

3.4.3 Load Splitting

This is an introductory paragraph to the material that follows in the subparagraphs. The data presented in this section of the handbook will give the reader guidelines for establishing partitioning of function and the number of busses. Once these guidelines are used, quantitative results can be obtained using computer aided design tools. Some of the anticipated guidelines include:

- o Subsystem-unique requirements
- o Communication requirements
- o Signal criticality
- o Level of redundancy
- o Processing requirement
- o Channel capacity (with respect to real-time)

These guidelines and the rationale for them will provide the system designer and systems software designer assistance to achieve a consistent approach to system architectural partitioning.

3.4.3.1 Dual Bus System

Advantages are discussed.

3.4.3.2 Multi-Bus System

Advantages are discussed.

3.4.3.3 Separate Buses

Advantages and current examples are discussed.

3.5 System Control Considerations

System control is related: (a) to the architecture considerations; (b) to the partitioning of the bus control hardware and software; (c) to the allocation of control among processing elements; and, (d) the allocated system performance requirements, i.e., what is expected of the integrated system of sensors, processors, controls, displays, etc. This section of the handbook will provide the appropriate introduction and synthesis of all system control considerations.

After this general discussion of system control, an introduction to the subtopics (bus control software, bus message protocol, error detection and correction) of the section will be presented.

3.5.1 Bus Controller Software

The requirements and source of information for the "table" in table driven bus control software will be presented and related to system control design.

3.5.2 Bus Message Protocol

This section will discuss various types of data traffic to be expected in an integration. Data traffic and the associated message characteristics are a function of the following system factors:

Mission Profiles,
Sensors, and
Application Software.

3.5.2.1 Synchronous and Asynchronous Messages

Message traffic associated with a typical system will be described in terms of the synchronous and asynchronous traffic it requires. The method for handling this traffic using the MIL-STD-1553 message formats will be detailed.

3.5.2.2 Abnormal Requests

In addition to the system data traffic, the multiplex data bus system requires certain messages to manage itself. These involve:

System event announcements - asynchronous.
Error test - synchronous and asynchronous.
Error recovery - asynchronous.

In this area the handbook will discuss the use of the mode codes, status bits, and error recovery procedures as they apply to the system integration.

3.5.3 Error Detection and Correction

3.5.3.1 Bit Error Rates

An analysis of bit error rates and bus utilization efficiency is required to support mission requirements for any weapons system. The effect of bit errors and an indication of acceptable error rates are discussed for several system functions and sensors.

3.5.3.2 Error Detection Theory

A simple analogy will be used to demonstrate the capability of the parity check, the check sum, and detection and correction codes. The detailed theory is referenced but specific performance examples are given with sufficient explanation of all errors to enable the reader to understand how the bit error rate for the standard data bus can be improved for critical data with minimum effect on bus efficiency.

3.5.3.3 Monitoring the MUX Bus State of Health

The extent of bus monitoring is dependent upon its purpose within the system. In the handbook, the reader will be apprised of current monitoring techniques available at present and their limitations.

3.5.4 Pre and Post Flight Checkout

This section of the handbook will describe how special system modes which can control the pre and post flight checkouts are created and incorporated into the system. The following tasks or modes are discussed:

a. Performance preflight go/no-go avionics check
b. Determine RT status
c. Record system errors for maintenance purposes

3.6 Bus Network Considerations

The substantial effect of bus network configurations on bus error rate performance will be treated in this section. In other words, error rate is a function of the system network configuration as well as bus interface hardware designs. However, bus error rate performance is not easily predicted for a specific bus network because of the many variables affecting the quality of a waveform from the time it is transmitted to the time it is received. These variables can be categorized as: (1) factors affecting waveform integrity at any given stub, and (2) characteristics of the receiver affecting data reception at the stub. The results of computer simulations will be presented to aid the system designer in getting insight into the influence of the variables.

The considerations in specifying the data bus wire and connectors will also be included in this section.

3.7 Multiplex System Design in the System Acquisition Life Cycle

The purpose of this section is to relate the multiplex system design considerations and decisions to a normal system acquisition.

3.7.1 Conceptual Phase

This paragraph will discuss the relationship of studies of proposed solutions of systems for satisfying a military need, required operational capability, etc. to integration using MIL-STD-1553. The level of detail

to which an initial system specification establishes and defines the role and method of integration via multiplex buses will be discussed.

### 3.7.2 Validation Phase

This is the period when major system characteristics are refined through studies, system engineering, and preliminary definition of the multiplex system. It may also include prototype or brassboard development, test, and evaluation. From these efforts, an authenticated system specification, and preliminary development specifications (both CI and CPCI) result.

This paragraph will relate the general way that studies of operational needs, system requirements (e.g., for data communication and transfer), and system optimization may affect the definition of a multiplex system.

### 3.7.3 Full Scale Development Phase

In a normal system acquisition, the extent of integration using the data bus is defined prior to the start of FSD, but many of the details of the integration remain to be defined and developed.

This paragraph will present the general chronology of FSD to multiplex system, hardware and software specification, development, and verification. The scope of formal PDRs and CDRs will be included, to provide quidance for the level of detail and the questions to be answered at those reviews.

The need for, and description of testing (laboratory, "hot bench", ground, and flight) as it relates to multiplex systems will be included.

## 4.0 HARDWARE DESIGN

### 4.1 Multiplex Terminal Functional Partitioning

This subsection will provide a generalized description of multiplex terminal functional elements and considerations for partitioning which apply to remote terminals, bus controllers and monitors. a brief description of remote terminals, bus controllers and monitors related to typical system operation will be presented for introduction to this section.

## 4.2 MIL-STD-1553 A/B Requirements Summary

This subsection will provide a summary and comparison of MIL-STD-1553 A/B requirements which have significant effect on the terminal hardware design, especially at the bus interface. Special consideration will be given the key characteristics of signal levels, waveform, noise and loading effects at the bus/terminal input/output port. The effects of stubbing from the trunk to the terminal will be briefly reviewed to show why external couplers may be required to maintain signal integrity and fault isolation characteristics of the bus network. A discussion of shielding and grounding techniques for EMI control will be presented. A detailed review of network configuration and effects on signal integrity is contained in Section III.

Test and evaluation criteria will also be considered in the summary.

## 4.3 Bus Coupler Characteristics

The key characteristics of the external bus coupler will be described in the following paragraphs

### 4.3.1 Transformer Characteristics

The following characteristics of the coupler transformer will be considered.

- o turns ratio
- o impedance
- o droop characteristic
- o shunt inductance
- o common mode rejection

### 4.3.2 Packaging

This section will describe the physical properties of various couplers. The important considerations include size, shielding, grounding and mounting provisions. In addition, the selection of connectors, if used, will be discussed.

## 4.4 Transceiver Characteristics

### 4.4.1 Overall Considerations

The general functional characteristics and the considerations for

transceiver design related to the Mil-Standard requirements will be listed and discussed.

4.4.2 Transmitter Design

The following areas will be considered for their impact on transmitter design and performance:

- o output signal waveform and power
- o voltage or current drive
- o output impedance
- o transformer characteristics
- o fault protection and control
- o noise
- o technology and packaging

4.4.3 Receiver Design

The major factors affecting receiver design are listed below and will be considered in this discussion:

- o filter selection/design
- o detection methods (analog/digital)
- o receiver threshold/noise/error rate
- o transformer characteristics
- o technology and packaging (existing and planned devicves)

4.5 Remote Terminal Design

4.5.1 RT Functional/Physical Partitioning

The remote terminal will be partitioned to identify the functions that must be performed in various classes of applications. The basic functions will be described and considerations for physical partitioning and implementation reviewed.

4.5.2 RT Design Examples

Examples of RT designs and implementations will be described. These examples will span the range of RT types from stand-alone with extensive subsystem signal conditioning to the transparent word processor interfacing with an intelligent subsystem processor.

4.6 Bus Controller Design

4.6.1 Controller Functional/Physical Partitioning

The controller functions will be identified and considerations for physical partitioning and implementation will be reviewed.

4.6.2 Controller Design Examples

Examples of controller designs varying from stand-alone and flexible (programmable) computer interface to simple bus interface (word processor) will be described.

4.7 Bus Monitor Design

4.7.1 Monitor Functional Partitioning

Monitor functions will be identified and possible combinations of these functions will be described.

4.7.2 Monitor Design Examples

Examples of monitor designs resulting from various operational systems requirements will be detailed.

4.8 Advanced Technology Implementation

A discussion of circuit and packaging technology available for multiplex terminal implementation will be described. Current state of the art devices and those expected to be available in the near future will be identified. An evaluation will be presented to show the possibilities for application of advanced hybrid and LSI technology to future terminal designs.

5.0 SOFTWARE ARCHITECTURE

5.1 System Control Philosophies

Several different system control philosophies will be outlined in this section (paragraph 5.1) to provide a basic set of concepts which can be compared with respect to the points outlined below. This section on control philosophies establishes the understanding for Sections 5.2 and 5.3 (Control of Application Software and Application Software Design Considerations).

### 5.1.1 Contrast Control Procedures with Requirements

Requirements on a data processing system such as redundant processing of data, cyclic frequency of updated data, allowable latency of data, etc. will affect the choice of a control philosophy. How the control mechanisms impact a variety of requirements and vice versa will be discussed.

### 5.1.2 System Synchronization

The synchronization of the software functions plays an important part in the control of most systems. Each control philosophy may have different mechanism whereby the software functions can be synchronized in a distributed system.

### 5.1.3 Allocation of Control

The function of "who is in control" and "how to establish the control" state, initially, and during any reconfiguration varies by control procedure and is a subject frequently glazed over by designers.

### 5.1.4 System Error Handling and Recovery

The mechanism for error recovery depends both on the control procedures and upon the requirements imposed upon the system. For example, triple redundant flight controls error handling would be significantly different from error handling for functionally redundant units. The control design has a significant impact on how these functions will be monitored for correct functioning and how recovery will occur once an error has been detected. The following subtopics will be discussed:

a. Mechanisms to monitor the multiplex system for proper functioning
b. Error detection
c. Error isolation
d. Error recovery techniques.

## 5.2 Control of Application Software in a Multiplex System

### 5.2.1 Master Executive Control Function

The executive functions are related to the multiplex control philosophy, and this will be explained. The importance of making the executive independent from the system control will be highlighted. Then the functions of the master executive will be discussed, organized as follows:

Since bus control is a function of the master executive, bus control software will be thoroughly described. The general method by which table driven bus control software is designed and the functions of message lists will be related to bus control software design.

5.2.2 Local Executive Control Function

The duties of a local executive in a processing element will be defined and discussed.

5.2.3 Relationship with System Control

The coupling of the control procedures with the executive functions will be explored and described. The function of the local executive with respect to system synchronization will be established here.

5.2.4 Method of Facilitating Applications Communications over the MUX

The specific schemes to allow the tasks of one processor to communicate and to control tasks in another processor will be discussed in conjunction with the desirability of allowing multi-computer task coordination.

5.2.5 Local Error Control

The errors which the local executive is tasked with recovering will be discussed. Techniques and impacts on the system error control will be expanded here from the discussion in system error handling and recovery.

5.3 Application Software Design Considerations

5.3.1 MUX Impact on Application Software

Application software organization and design is impacted because of the physical distribution of functions over multiple processors and the information transfer system capabilities and restrictions. The relationship between the multiplexing of information transfers and the application software will be discussed.

5.3.2 Independence of Modules

A design goal of all software is to construct modules which are maximally independent. The greater importance of module independence will be described as it relates to the desirable goal of being able to move a module from one processing element to another to redistribute the

processing load for a distributed processing system. The general techniques for contruction of software will referenced in this section.

5.4 Support Software

While most of the support software is common between a multiplex system and any other computer system, communication via messages between one precessing element and another will have an impact on the support software. Synchronous messages, their frequency, source and destination, processor location and impact of arrival of the message can be automated for incorporation into the total system. The effects and utility of specialized tools to facilitate software development will be described.

6.0 MULTIPLEX SYSTEM EXAMPLES

The candidate systems that will be described in this section have been chosen:

B-52 Offensive Avionics System
F-16 Avionics
F-18 Avionics
LAMPS
Advanced Attack Helicopter
Digital Avionics Integration System

Final selection will be based on the following criteria:

- o Current state-of-the-art application in aircraft subsystems.
- o Utilization of MIL-STD-1553 or very similar requirements.
- o Long term system utilization which requires extensive retrofit.
- o Availability of system design details, including rationale for characteristics selection.
- o Variety of subsystem applications (AMUX, EMUX, Stores, management, Flight control, communications control, etc.)

The following is a preliminary approach to the major subjects to be discussed in this section with a listing of information content.

6.1 Selection of Examples

- o Criteria for selection
- o Information sources
- o Listing of examples/brief description

6.2 - 6.7 Examples #1 Description

(All example paragraph headings are identical to those given below:

6.X.1 Application Area

- o (EMUX, AMUX, Stores Management, Flight Controls. etc.)
- o Extended application
- o Type of information/control signals transmitted
- o Signals (Parameters)
- o Subsystem equipments
- o Unique requirements

6.X.2 System Architecture

- o Physical/functional
- o Documentation/comparison to MIL-STD-1553A/B
- o Deviations from MIL-STD
- o Redundancy level/type
- o Bus network/length/stubbing/couplers/limitations
- o Synchronization method
- o Bus protocol
- o Message size

6.X.3 System Control

- o Fault isolation and reconfiguration
- o Primary/backup control
- o BITE
- o Software
- o Unique modes

6.X.4 Bus Controller

- o Primary/backup
- o Software/firmware control
- o Flexibility
- o Controller subsystem type
- o Self test
- o Reconfiguration

6.X.5 Remote Terminal

- o Locations/types

- o Integral to subsystem or standalone
- o Subsystem interface
- o Redundancy level/reconfiguration
- o Control of faulty transmitter
- o Self test and fault isolation
- o Size/weight/power
- o Device technology
- o Unique features

6.X.6 Rationale for System Design

- o Hardware
- o Software
- o Network/couplers/stubbing
- o Signals included/excluded
- o Redundancy level
- o Waveform
- o Grounding/shielding
- o BITE level
- o ERROR rate tolerance
- o Deviations/additions/deletions from MIL-STD

# A COMPARISON OF TIME DIVISION MULTIPLEX DATA BUS TECHNIQUES BASED ON DATA TRANSFER PERFORMANCE

Charles R. Husbands

The MITRE Corporation *
Bedford, Massachusetts

## ABSTRACT

This paper compares the performance of three time division multiplex bus techniques based on data transfer efficiency. The sensitivity of the data handling capability of each of these multiplex bus techniques to a series of parameter variations is examined. These parameters include bus operating frequency, data message length, the interword and intermessage gaps between transmissions, and the number of active terminals employed in the bus system. The results of this parametric examination describe the upper bound of data transfer performance for each of the TDM bus techniques.

## INTRODUCTION

The purpose of this paper is to present the results of a series of parametric studies designed to investigate the upper bound on data transfer performance for several time division multiplex bus techniques. To accomplish this task, three bus protocol techniques were examined. Each of the selected bus techniques examined in this study is either currently in use, or is being considered as a candidate for application to military avionics information distribution systems.

The first bus arbitration technique examined uses the command-response protocol as described in MIL-STD-1553A (Reference 1). This technique and standard has been extensively employed in a series of Air Force Avionics Programs. The second system examined investigates a data bus constrained to the Polled-Contention arbitration technique as described in MIL-G-85013 (AS) (Reference 2). This format and protocol was developed by the Navy Air Development Center and is normally referred to as the General Purpose Multiplex System (GPMS) format. The GPMS architecture has been examined for military applications in both shipboard and avionics systems. The third bus arbitration technique examined is a pure contention protocol, and is implemented in a system configuration called the Listen-While-Talk System. The Listen-While-Talk (LWT) technique, under investigation at The MITRE Corporation, has been applied to several ground-based information transfer systems

* This paper released under the sponsorship of the Electronics Systems Division of the Air Force Systems Command.



PRECEDING PAGE BLANK

and is currently being examined as a possible candidat for an Air Force Avionics Bus Program.

In this paper, the upper bound on bit transfer rates and information transfer efficiency will be examined under the conditions imposed by the applicable standards, and then these results will be examined under conditions where some of the constraints imposed by those standards were modified. The interest in evaluating modifications to the existing parameters is driven by the possible use of fiber optics. Existing standards call for the use of twisted shielded pair cable as the transmission medium and this factor limits the bus operating frequency to the 1 Megahertz region. The use of fiber optics removes that data rate restriction and bus operation frequencies up to 10 Megahertz has already been demonstrated (Reference 3). In this report, the data transfer capability of the system was examined at data rates between 1 Megahertz and 10 Megahertz. The sensitivity of data transfer capability to changes in the interword gap was investigated, as well as the effects of variations in interword gap. An examination of variations in interword gap on data transfer efficiency and information transfer capacity over the range of 2 to 10 microseconds was also included in this study. The system characteristics and data transfer performance of the three selected systems were compared. This comparison indicates the sensitivity of each of the protocol techniques to variations in bus parameters and describes the advantages of one bus technique over the others in areas where this is applicable.

## MIL-STD-1553A CONFIGURATION

The document MIL-STD-1553A describes a command-response bus protocol designed to allow internal communications of digital messages within an aircraft. The basic Command-Response architecture, described in MIL-STD-1553A, is shown in Figure 1. Two types of terminal units are available in this configuration; remote terminals and the bus control interface terminal. At any one time there is only one active bus controller and all other terminals are considered remote terminals.

In this command-response configuration all commands originate at the bus controller. The bus controller can initiate three types of commands: Remote Terminal to Control Processor Data Transfers, Control Processor to Remote Terminal Data Transfers and Remote Terminal to Remote Terminal Data Transfers. It should be noted that the Remote Terminal to Control Processor and Control Processor to Remote Terminal Modes require the same number of overhead words to achieve each data transfer, so only one of these modes will be analyzed in this paper. This mode will be abbreviated as the CU/RT (Control Unit to Remote Terminal) mode. The other mode to be analyzed is the Remote Terminal to Remote Terminal mode and will be abbreviated as the RT/RT mode. The RT/RT mode is a matched pair of CU/RT transfers. The controller first requests the receiving remote terminal to listen and then asks the sending remote terminal to transmit. During the exchange of data, both remote terminals respond with a status word at the proper time.

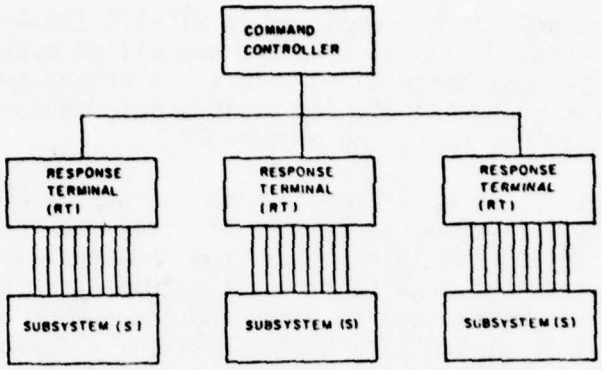



Figure 1 Typical Command-Response Architecture

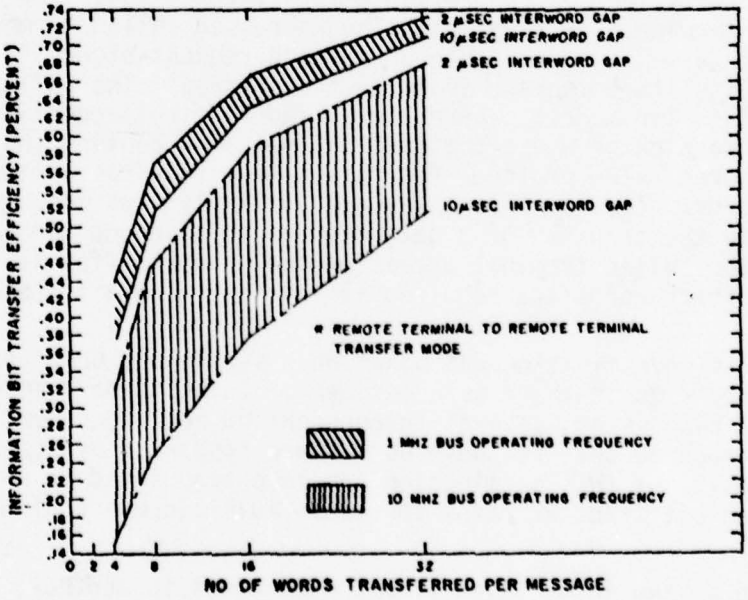



Figure 2 Effects of Parameter Variations on Information Transfer Efficiency (Command-Response Protocol)

In the word format employed in MIL-STD-1553A with the transmission of each data word, 3 bits of sync and one bit of parity is transmitted. These bits must be considered along with the overhead produced by the command and status words when establishing maximum information data transfer rates and the effective bus operating efficiency.

A plot of the upper bound on data transfer efficiency for a bus constrained to this command-response protocol and format is shown in Figure 2. In this diagram, the information transfer sensitivity of the remote terminal to remote terminal transfer mode is illustrated for various values of interword gap, bus operating frequency and the number of words transferred per message. This figure illustrates that for a fixed number of words transferred per message, the information transfer efficiency will decrease for increasing values of interword gap and bus operating frequency.

## NAVY GPMS CONFIGURATION

A typical polled contention architecture of the type used in GPMS is shown in Figure 3. For the analysis done in this report, only a one bus configuration was considered. In a one bus configuration the system components would consist of the data bus, a polling controller and a series of polled terminals. The polling controller acts only as an independent bus arbitrator. The bus controller offers the available bus to each interested polled terminal, in turn, with the addressed polled terminal accepting the bus with a message available or message request signal if interested, or not responding if no message is to be transferred. The polled contention system can act in two modes: where the polled terminal accepting the channel offer is a data sink or where the polled terminal accepting the channel offer is a data source. The protocol for this system requires a minimum of three overhead words: the bus offer, message available, and message request, to initiate the transfer of a data message to a second terminal. In the case where the polled terminal accepting the channel offer is a data sink, only two overhead words are required to initiate a data transfer.

This investigation was based on a single bus GPMS configuration, where for every request there is a response. In a standard mechanization of this type of bus system, several independent busses may be employed to provide active redundancy. If these busses are supported by non-blocking terminals, the results of this examination can be extrapolated to establish the upper bound on bit transfer rates for these multiple bus configurations.

In moving information from one terminal to another, the upper bound or data transfer performance will be established by the way the data is blocked, the overhead associated with each data word transfer and the overhead required to establish message communications. The data word format used in this polled-contention mechanization is identical to the data word employed in MIL-STD-1553A.

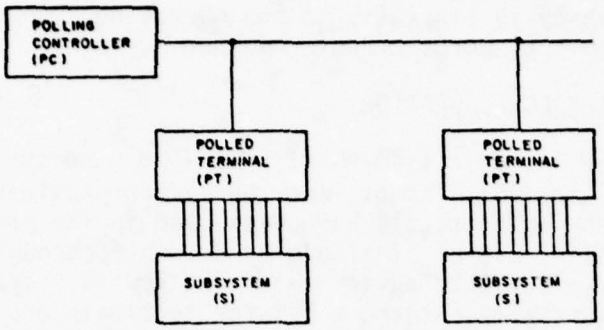



Figure 3 Typical Polled-Contention Architecture

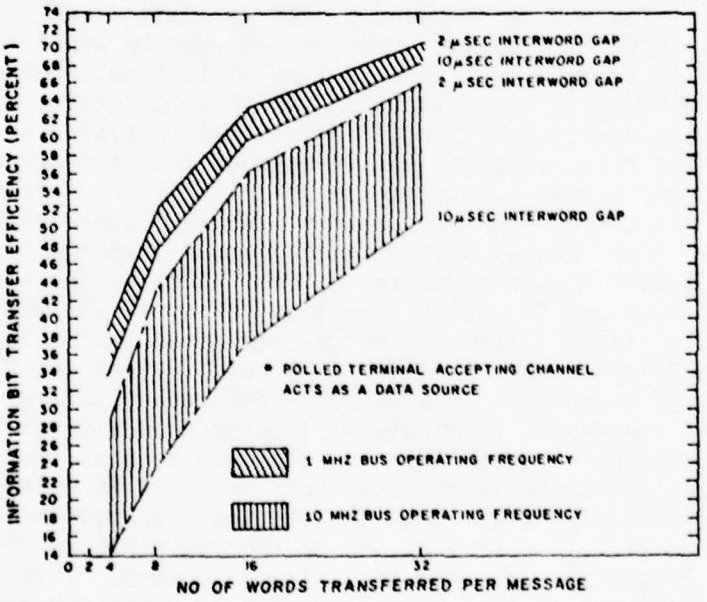



Figure 4 Effect of Parameter Variations on Information Transfer Efficiency (Polled-Contention Protocol)

The effects of parameter variations on information transfer efficiency, for the mode where the terminal accepting the bus offer is a data source, is shown in Figure 4. It can be seen from this figure that information transfer efficiency is sensitive to bus operating frequency, the interword gap and the number of words transferred per message.

## LISTEN-WHILE-TALK CONFIGURATION

The Listen-While-Talk system, or LWT, is a pure contention system utilizing no central control element, and the system provides distributed network control. Contention protocols have been used in the past in an effort to make more efficient use of available bandwidth for computer and terminal communications. They take advantage of the low duty cycle or the "bursty" nature of data normally exchanged between terminals and computers. Given a large number of bursty users, the contention protocol uses the law of large numbers to provide sufficient bandwidth to match the average aggregate data transmission rate of the entire population, rather than matching the peak rate of each device.

The LWT contention technique developed at The MITRE Corporation is an outgrowth of the ALOHA System developed at the University of Hawaii, and the Ethernet System developed by the Xerox Corporation.

The LWT contention system, as currently mechanized at The MITRE Corporation is shown in Figure 5. The system consists of an "outbound cable" for data transmission and an "inbound cable" for data reception. This unidirectional concept allows the receiver at each terminal to listen to all other possible transmissions on the line while that terminal is transmitting. A terminal wishing to transmit first monitors the status of the line. If no other units are transmitting, the terminal initiates transmission. Because of the relative positions of various transmitters and receivers in the system, two terminals may decide to initiate transmissions almost simultaneously, and the propagation delay of the line would allow each to have an essentially clear channel at the initiation of transmission; hence the Listen-While-Talk requirement. After initiation of the transmission, the receiver is required to monitor the line for conflicts with other transmitters for at least a time period corresponding to the longest possible propagation delay in the system. If no conflicts occur, transmission continues. If a conflict occurs, both conflicting terminals will observe the phenomena, and both terminals will cease transmission. Since there is no central control point to arbitrate who should be the first to again initiate transmission, the transmission from each terminal is delayed for a random time interval and then retried. Because the transmissions are terminated as soon as a conflict is observed, the maximum possible conflict time will not exceed the longest propagation delay in the system. This rapid abort on sensing conflicting transmissions causes only a minimum amount of data to be lost due to data collisions on the line.

If a terminal wishes to transmit and the line is in use at the beginning of the scheduled transmission, the transmission will be delayed until the

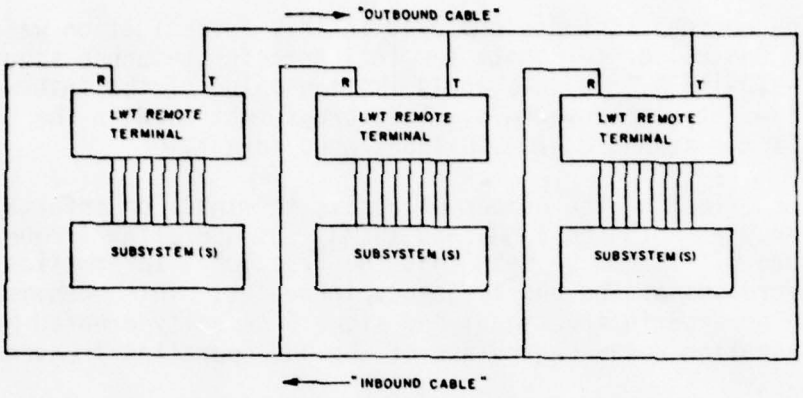



Figure 5 Typical Listen-While-Talk Bus Architecture

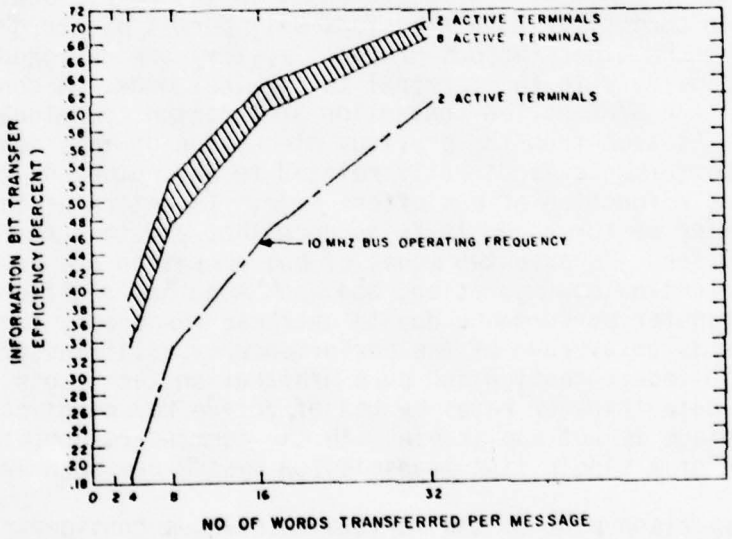



Figure 6 Effect of the Number of Active Terminals on Information Transfer Efficiency

line becomes free. At that point, the terminal will compete for the line with the other terminals requesting the medium according to the process just described.

The message sequence employed in this investigation was taken from the Command Controller to Remote Terminal transfer response sequence described in MIL-STD-1553A. In this analysis, the value of the interword gap (T) has been fixed at 5 microseconds. The word format used in the Listen-While-Talk model is the standard MIL-STD-1553A word format.

The effect of the number of active terminals on information transfer efficiency at both the 1 $MH_z$ and 10 $MH_z$ bus operating frequencies is shown in Figure 6. It can be seen that the fractional information capacity of the line decreases as the bus frequency increases. This phenomena occurs because of the increase in fractional "no signal" capacity created by the fixed value of propagation delay regardless of the bus operating frequency.

## COMPARISON OF BUS ARBITRATION TECHNIQUES

In order to allow a comparison of data transfer performance for the various bus arbitration techniques covered in this report, a series of assumptions had to be made. These assumptions were imposed to allow the three systems to operate under the same set of constraints. For the MIL-STD-1553A command-response protocol, only the results of operation in the terminal to terminal mode were presented in the comparison. Although remote terminal to controller data transfers will permit higher data rate transfer capability, the other two bus protocol systems are designed to transfer information exclusively in the terminal to terminal mode. A constraint was also imposed on the GPMS polled-contention arbitration technique for comparison purposes. As seen from the previous discussion of this technique, data transfer performance is directly related to the number of terminal responses received as a function of bus offers made. To determine the upper bound on data transfer performance, it is assumed that a data transfer will occur for each bus offer. Because two modes of bus operation are possible with the polled-contention configuration, and each mode has a different upper bound on data transfer performance due to overhead responses, the data used for comparison is an average of the performance capabilities of the two modes. Both the polled-contention and pure arbitration techniques can increase their effective data transfer rates by use of active redundant cables. Because this technique is not applicable with the command-response protocol, the assumption of a single line transmission system has been made.

A comparison plot of the various techniques considering the assumptions just described is shown in Figure 7. This figure describes the upper bound on data transfer rates for the three bus arbitration techniques operating at a one megahertz bus rate. From this diagram it can be seen that the Listen-While-Talk technique demonstrates the best data transfer rate performance for a 2 terminal system, but as the number of active terminals increases, the data transfer rate decreases. This decrease, as previously discussed, results from increased overhead created by collisions. Figure 8 shows a

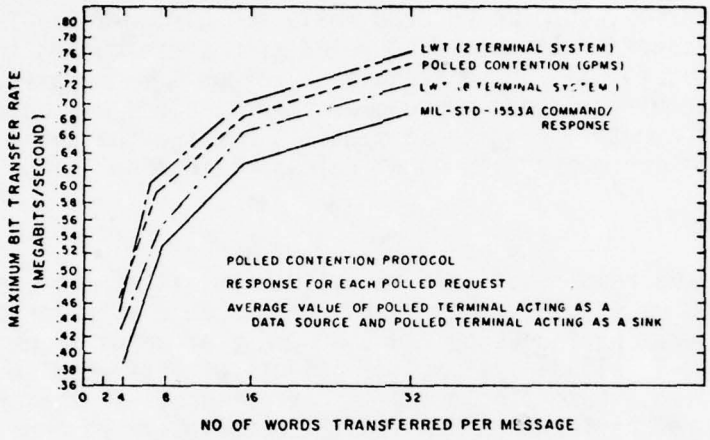



Figure 7 A comparison of the Upper Bound on Bit Transfer Rate Performance at 1 Megahertz

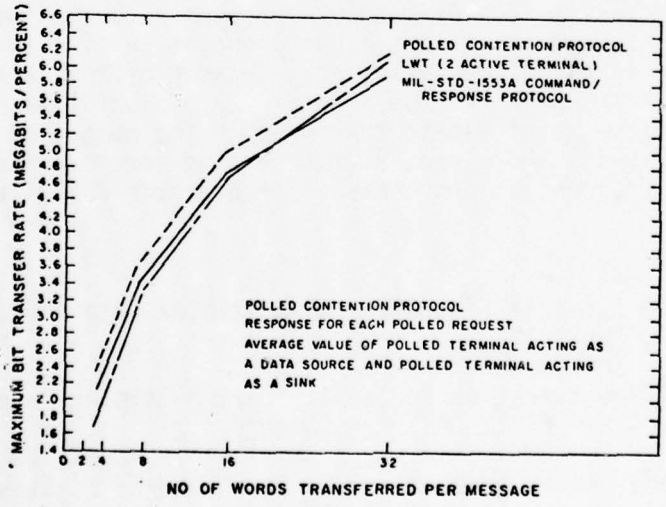



Figure 8 A Comparison of the Upper Bound on Bit Transfer Rate Performance at 10 Megahertz

plot of the same information for a bus operating at 10 megahertz. It can be seen from this figure, that with only 2 active terminals, the Listen-While-Talk data transfer performance is less than the polled-contention system, and performance is comparable to the command-response system operating in the remote terminal to remote terminal mode. In both Figures 7 and 8 the interword and intermessage gap was fixed at 5 microseconds. As shown earlier, when the performance results of the individual techniques were examined, the command-response and polled-contention techniques are sensitive to interword gap. If the interword gap was increased from 5 to 10 microseconds, the data transfer rates for the polled-contention and command-response curves would be reduced while the Listen-While-Talk curves would remain unchanged. It should be noted from these curves that data transfer performance, for all three techniques, drops off dramatically as the number of words transferred per message decreases. This has become a major problem in the implementation of these systems, because the average number of words transferred per message in an avionics application tends to be very small.

## CONCLUSIONS

From the results of this investigation, it can be seen that for all the bus techniques studied, the upper bound on data transfer performance is sensitive to operating frequency and the number of words transferred per message. For the MIL-STD-1553A system, the effects of interword and intermessage gap also affected information transfer performance. In the GPMS technique the data transfer performance was sensitive to interword and intermessage gap, and the number of active terminals in the network. Under conditions where a fixed response was solicated from each terminal interrogated, this last parameter has no effect on the upper bound on data transfer performance. The number of terminals in the network responding to the polling controller does affect the transfer capability between terminals for long message exchanges, i.e., greater than 32 words. The Listen-While-Talk technique, while not suffering from decreased performance due to interword or intermessage gap, is sensitive to the bus propagation time and the speed of detecting conflicting transmissions. Because of the lack of a control arbitrator in the network, the probability of conflicts increases with the number of active terminals in the system. The number of active terminals in the network is, therefore, a constraining parameter in establishing the upper bound on transfer performance for a Listen-While-Talk System.

## REFERENCES


1. "Aircraft Internal Time Division Multiplex Data Bus", Military Standard 1553A, 30 April 1975.

2. "General Specification for an Aircraft Multiplex Data Bus", Military Standard MIL-G-85013 (AS) Preliminary.

3. "Optoelectronic Data Bus", J. R. Baird and L. L. Steward, Spectronics Inc., AFAL-TR-73-271, October 1973.

MULTIPLEX DATA BUS CONFERENCE

10-12 OCTOBER 1978

DAYTON, OHIO

SESSION V

# DATA BUS HARDWARE

MODERATOR: Colonel Clifford H. Allen
AFATL/DLM
Eglin AFB, FL

# USE OF STANDARD MODULES FOR AIRBORNE MULTIPLEXING APPLICATIONS "CHAPTER 2"

J. W. McCuen
G. E. Smith

Hughes Aircraft Company
Microelectronic Products Division
Newport Beach, California

This paper is an update of one given at the 1st Multiplex Data Bus Conference, in November of 1976. It describes the LSI/Hybridized MIL-STD-1553 Remote Terminal in development at the Microelectronic Systems Division of Hughes Aircraft Company. The design concept is reviewed along with those changes necessary to incorproate the 1553B requirements. The paper discusses several interfaces and programming functions which adapt the Remote Terminal to various MIL-STD-1553 applications, e.g., F-16, F-18 aircraft and 1553B MUX systems.

PRECEDING PAGE BLANK

## INTRODUCTION

The Hughes MIL-STD-1553 Remote Terminal (RT) design concept evolved from a problem faced by many large companies that are developing avionics hardware. If the company has many divisions, each producing a variety of avionics and missiles systems, each division will in certain common hardware areas tend to re-invent the wheel. This is true of multiplex hardware required of these avionic and missile systems. This problem is magnified when a particular avionics system is required to operate on different type aircraft or weapon systems that have different type MIL-STD-1553 systems, e.g., F-16 (1553 Basic), F-18 (1553A), new vehicles (1553B).

To solve the above stated problem, the Microelectronic Systems Division of Hughes accepted the task to develop LSI and hybrid devices that can be assembled into Remote Terminals which then can be modularized and programmed to meet the various 1553 applications. A study was conducted to determine the feasibility and economics of developing a universal RT design to achieve standardization and interchangeability of hardware. From the study the following listed design goals were established, against which each facet and parameter of the design was evaluated:

1. The RT hardware (Front End) must be adaptable to both single and dual bus systems.

2. Redundancy in the RT (between bus and subsystem) incorporated to the maximum depth possible. In a dual bus RT no single failure will disable the communication function of the terminal.

3. A standardized RT to subsystem interface established which will work with all type air vehicle subsystems without redesign of the RT Front End.

4. The RT must be programmable to work with existing 1553 MUX systems and with new 1553B applications.

5. The RT must be self supporting in that it requires no support from its host subsystem other than prime power forms.

6. The RT elements must be productized in LSI and hybrid technology for optimum adaptation (size and power) to all subsystem black boxes.

7. The RT and its functions must be simplified to the fullest extent.

   A. MUX bus synchronous word exchange with the subsystem, i.e., no message block memory.

   B. Standard (intermediate) interface to work with all subsystems.

These design goals have been met in the development of the RT and its elements. The following material describes and discusses the basic design concept of the RT approach.

## DESIGN CONCEPT

The Hughes Remote Terminal (RT) can best be described by reference to Figure 1 and 2 accompanied with description of the functional interfaces. Figure 1 depicts a representative dual bus RT showing the five interfaces involved in the operation of the RT. Figure 2 shows a singular Front End Module (FEM) which is the prime element of the RT.

### 1. Remote Terminal To Mux Bus Interface

This interface, for each RT application, is defined by the various vehicle MUX specifications (i.e., 1553) and therefore is not discussed here.

### 2. Remote Terminal to Subsystem Interface

This interface is, and will be for sometime to come, different with each application of a RT to different type subsystems. Hughes design concept does simplify the implementation of this interface by establishing an Intermediate Subsystem Interface between the RT Front End and the host subsystem wherein all necessary functions are provided for adaptation to the subsystem by a minimum of additional circuitry. This approach eliminates the differences in subsystem functions from reflecting back into the RT past the intermediate interface.

### 3. Intermediate Subsystem Interface (Standard)

In the Hughes design this interface is most critical since it affects standardization in usage of the RT. At this interface (See Figure 3) all known and necessary functions are present to enable any Avionics type subsystem to be serviced via bi-directional data transfer or an Electrical/Airframe type RT to control the programming of its I/O signal conditioning circuitry.

This interface, with the Subsystem Interface Logic (SSIL) cirucitry, buffers the ever changing function, required of the different host subsystems, from advancing back into the RT causing redesign of the FEM. The interface functions, in the majority of application (dual bus), will be hardwire OR-ed from the two FEM's to the appropriate SSIL. The interface functions are shown in Figure 3. An example is given of a SSIL and those intermediate interface functions used to provide bi-directional data exchange with an avionics communication subsystem (via DMA) which incorporates a microprocessor. Other interface functions labeled 'Provisional' are primarily used for implementing a RT which is used to control the processing of EMUX type signals, e.g, analog, discrete and synchro. Of particular interest are the

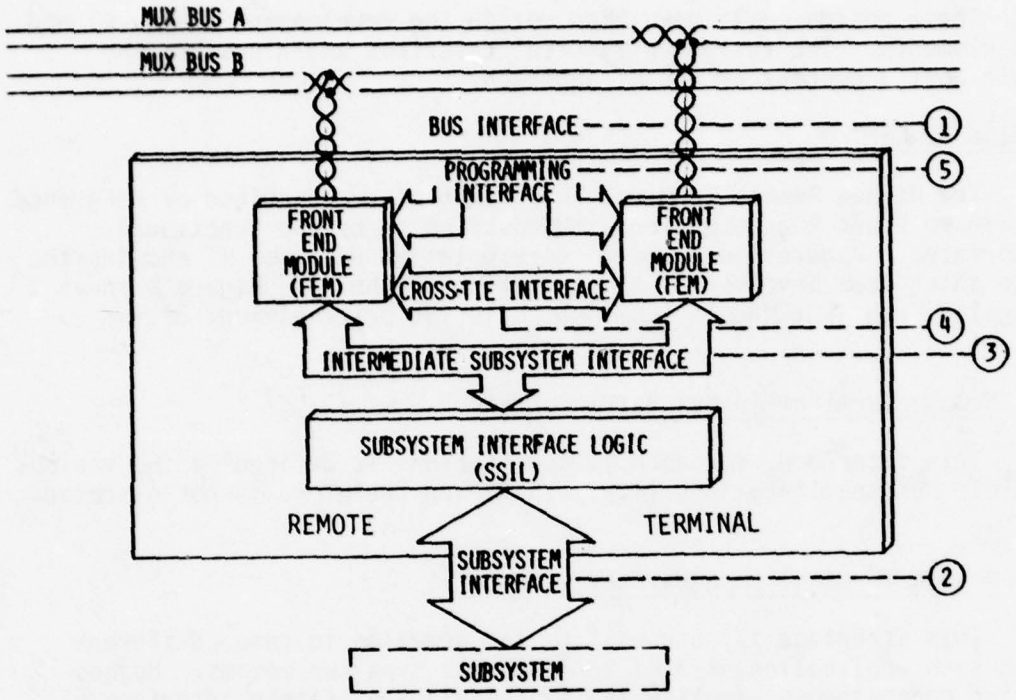



Figure 1. Representative Remote Terminal (Dual Bus)

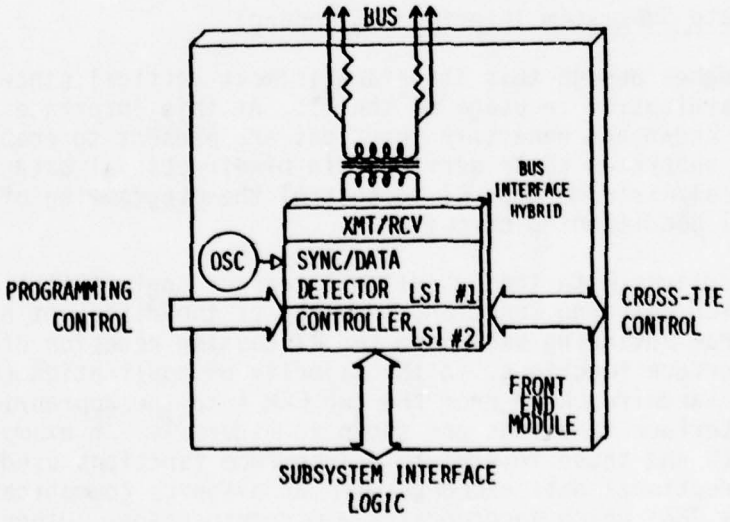



Figure 2. Representative Front End Module (Single Bus)

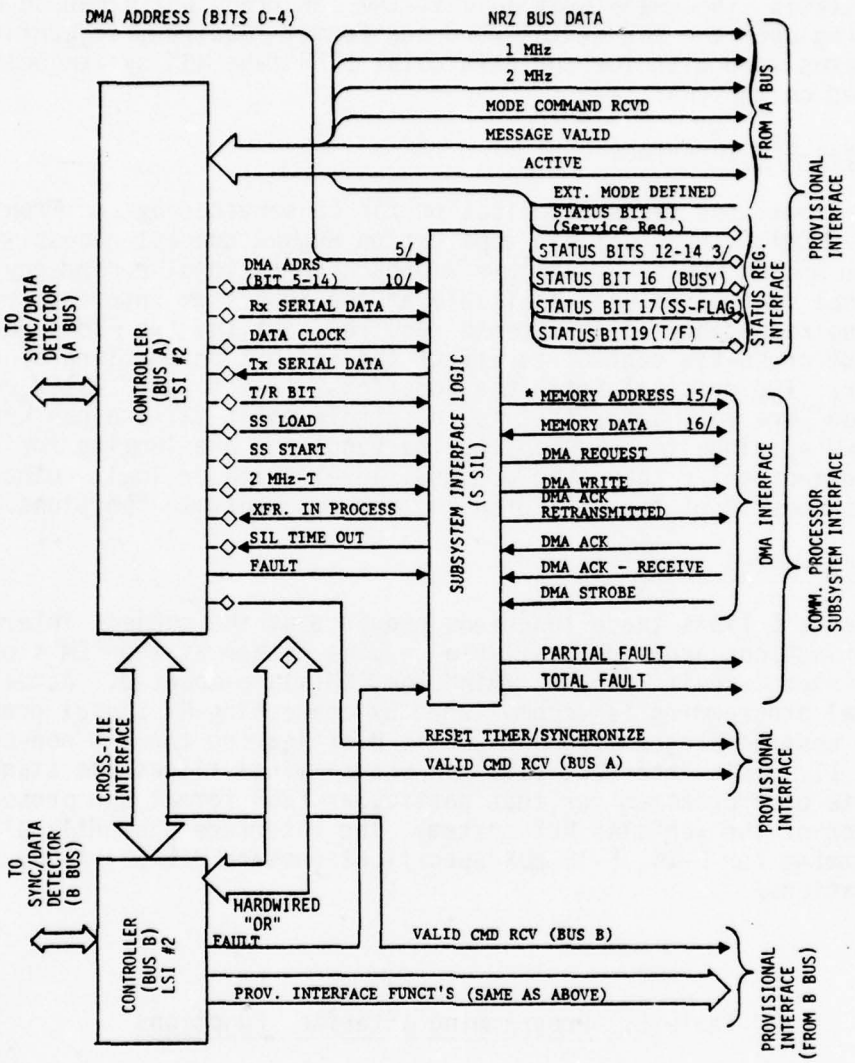



Figure 3. Intermediate Subsystem Interface

seven status lines made available at the interface which can be used depending upon the MUX status word and format required, to configure the status word bits for the particular 1553 type MUX system being employed on the vehicle.

4. Cross-Tie Interface

A single bus system application can be serviced by one Front End Module (FEM). In a dual bus application Hughes concept necessitates the use of two FEM's. With this approach operational redundancy and RT sectional control is made available at the subsystem interface. Carrying redundancy to this depth requires that the two FEM's have a means of cross-tie control to affect the inhibit or enabling of one another. Two critical inter-tie functions, 'reset' and 'valid command received' are fail safe, that is, no single point failure can knock out both FEM's, since the FEM's using the functions are looking for an edge (edge detection) rather than a signal level (high or low). Other signals consist of transmit disable, and bus shutdown functions.

5. Programming Interface

Table 1 lists those functions provided at the subject interface. These functions are made available on pins either at the FEM's or the Printed Circuit Board on which the FEM(s)are mounted. Actual physical programming is accomplished by connecting RT signal ground to the desired programming pin (Logic 0)or leaving the pin non-connected (Logic 1). This interface (via pin programming) allows the standard FEM's to be programmed for that particular 1553 format and protocol required of the vehicles MUX system. The interface presently allows programming for F-16, F-18 MUX Specifications and MIL-STD-1553B Standard applications.

Table 1. Programming Interface Functions

- o Bus Application (Protocol) Select (2 pins)
- o Broadcast Disable (1 pin)
- o Mode Control Select (1 pin)
- o RT-to-RT Response Time Select (1 pin)
- o Up/Down Word Count Select (1 pin)
- o RT Address (5 pins)
- o DMA Address Select (5 pins)

To explain the programmable functions one must first understand the internal functions performed by the Controller (LSI #2) that receives the programming inputs. The Controller incorporates a 30 line micro-programmed control bus, and performs the following functions:

- o Message Validation
- o Message Decoding
  - Address Decoding
  - Transmit/Receive Decoding
  - Subaddress/Mode Control Decoding
  - Mode Code Decoding
  - Word Count Decoding
- o Status and Data Encoding
  - Sync Pattern Generation
  - Status Word Generation and Formatting
  - Manchester Encoding
  - Subsystem Addressing
  - Fail-Safe Time-Out
  - Redundancy Interface Control
  - Data Word Transfer Control
  - Status Response Control

These programmable functions are implmented in the design of the Controller (LSI #2), a discussion of each follows.

- o Bus Application (Protocol) Select - Two binary coded pin inputs are provided for selecting the protocol that is used for one of the three applications, F-16, F-18 and 1553B. Differences in protocol occur in the area of status bit definition (see Table 2), resetting of status bits, data handling under error conditions, and mode code assignments (see Table 3).

- o Broadcast Disable - Provided for 1553B applications, one input allows the broadcast function of the terminal to be completely disabled, if desired.

- o Mode Control Select - One input is provided to define the mode control field either as 00000 or 11111 for terminal application.

- o RT-to-RT Response Time Select - One input selects the maximum time interval (either 7.5 us or 12.5 us) the receiving RT will wait for a status response from the transmitting RT once a RT to RT command sequence has been detected. This function is incorporated to facilitate the difference in RT status response time between the various MUX specifications.

- o Up/Down Word Count Select - One input determines if the message word count field controller output to the subsystem should be incremented or decremented by one for each data word received/transmitted. DMA addressing of the host subsystem memory may use an incremented count, while other applications use a decremented count.

- RT Address - Five inputs are provided for the selection of the unique RT address.
- DMA Address Select - Five inputs are provided for block message addressing of the host subsystem memory when DMA is used.

Table 2. Status Bit Allocation

| Bit No. | F-16 | F-18 | 1553B |
|---|---|---|---|
| 9 | Parity Error | Msg. Error | Msg. Error |
| 10 | Inst. | ND | Inst. |
| 11 | Msg. Error | ND | Serv. Req. |
| 12 | NA | ND | ND |
| 13 | NA | ND | ND |
| 14 | NA | ND | ND |
| 15 | Broadcast CMD RCV | ND | Broadcast CMD RCV |
| 16 | Mode CMD RCV | ND | Busy |
| 17 | Bus B Shutdown | ND | SS Flag |
| 18 | Bus A Shutdown | ND | Dyn Bus Control |
| 19 | RT Status | RT Flag | RT Flag |

ND - Not Defined NA - Not Applicable

Table 3. Mode Command Allocation

| CMD | F-16 | F-18 | 1553B |
|---|---|---|---|
| XMT Status | X | ND | X |
| XMT'R Disable | NR | ND | X |
| XMT'R Enable | NR | ND | X |
| Reset RT | NR | ND | X |
| Synchronize | NR | ND | X |
| Reset Timer | X | ND | NR |
| Inhibit T/F Flag | NR | ND | X |
| Override T/F Flag | NR | ND | X |

ND - Not Defined NR - Not Required

## ADAPTABILITY BY COMPONENT REVISION

One of the most difficult requirements in the design of a standardized RT applies to the generation of the RT transmitted waveforms required of different MUX specifications, e.g., sinewave type with limited harmonic content-vs-trapezoidal waves. Figures 4a, b, c depict the RT transmitted waveform limit envelopes for the F-18, F-16 aircraft and 1553B MUX system applications. Figure 5a shows the current Hughes RT baseline waveform designed toward the F-18 application. Figure 6 is a graph showing the variation in harmonic content between the MDC A3818 Spec. and the Hughes baseline. The graph shows that the Hughes waveform adheres to the harmonic content suppression of MDC A3818 except for a 3 dB variation at 3 MHz. Hughes studies and test indicate the Hughes baseline waveform is a optimized trade-off between harmonic suppression per MDC A3818 and circuit complexity required for practical implementation of a 1553 transmitter in a hybrid package. For F-16 and 1553B MUX system applications which require trapezoidal transmitted waveforms (see Figures 4b and 4c), the removal of six discrete components from the Front End Hybrid (transmitter section) produces the waveform shown in Figure 5b. The Hughes square wave option waveform falls within both the F-16 and the 1553B envelope limits.

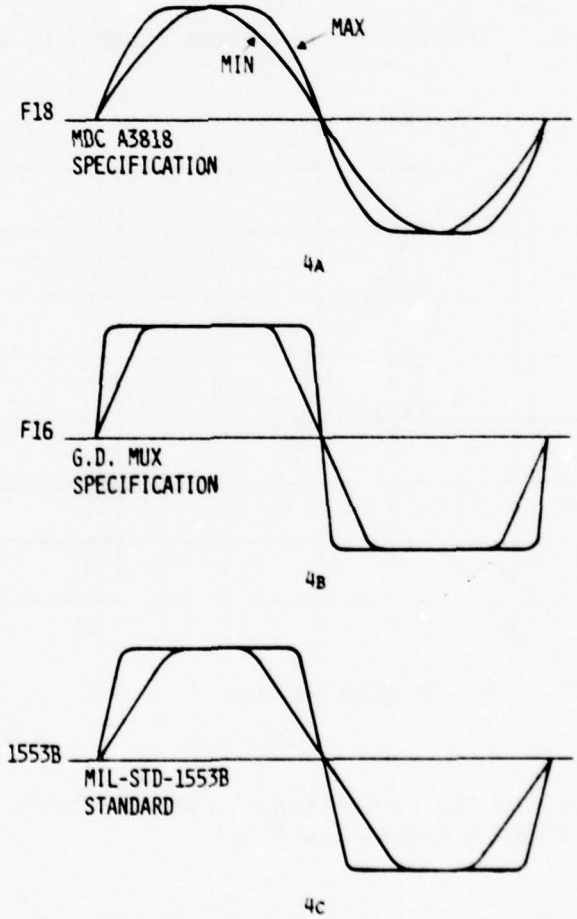
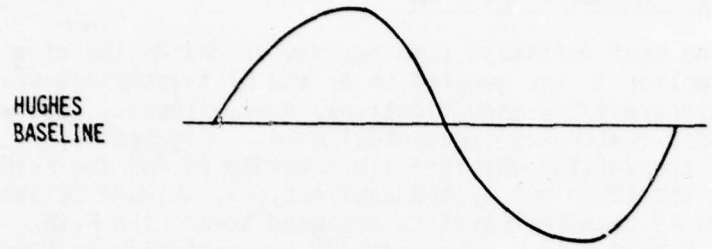
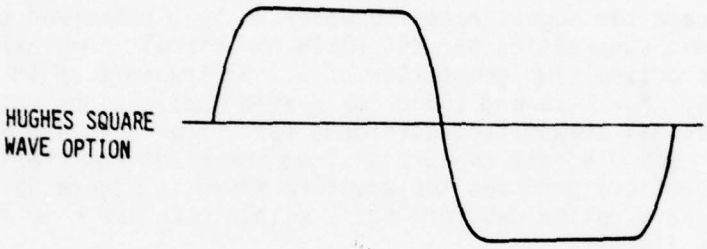



Figure 4(A, B, C). Transmitted Waveform Limit Envelopes




5A




5B

Figure 5 (A, B). Transmitter Waveforms (Hughes RT Concept)

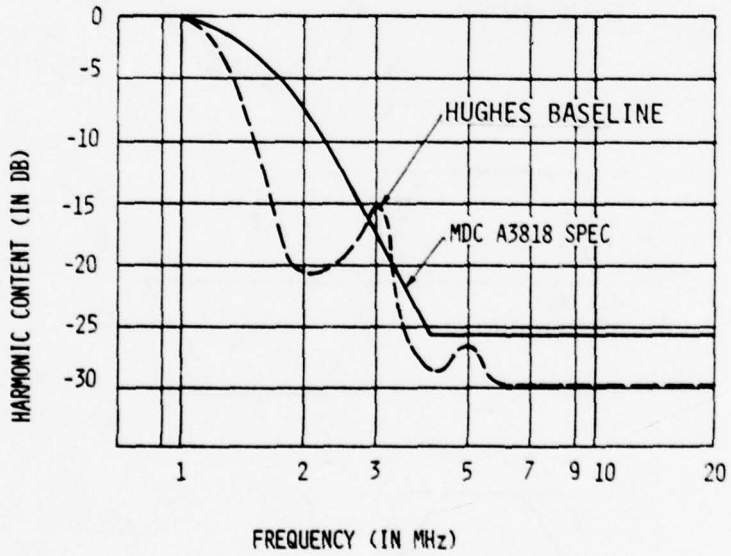



Figure 6. Comparison of Transmitted Waveform Harmonic Content (MDC A3828 & Hughes Baseline)

## GENERAL RT DESIGN PARAMETERS

- Radiation Level - RT circuit elements are designed to radiation levels of $10^4$ rads total dose and $10^8$ rad/sec. transient.
- Environmental Level - RT circuits designed to meet requirements of MIL-E-5400 Class II.
- Size/Volume/Power of RT Elements

### Front End Hybrid

| | |
|---|---|
| Size: | 2.4" x 1.6" |
| No. of pins: | 90 |
| Power Dissipation: | 4 watts |
| Circuit Technology: | |
| TX/RX - Discrete devices | |
| Sync/Data Detector LSI - SOS CMOS | |
| Controller LSI - Standard CMOS | |

### Subsystem Interface Hybrid

| | |
|---|---|
| Size: | 1.9" x 1.6" |
| No. of pins: | 72 |
| Power Dissipation: | 5 watts |
| Circuit Technology: | |
| Subsystem Interface Logic - 35 Schottky TTL, SSI and MSI devices | |

### Dual Channel RT with Subsystem Interface Logic

| | |
|---|---|
| Size: | 5.6" x 5.8" Printed Circuit Board |
| Power Dissipation: | 13.6 watts (20:1 transmit duty cycle) |
| Power Forms: | +5 Vdc |
| | ±12 Vdc |

# DUAL CHANNEL 1553A INTERFACE

Robert Hoffman

Applied Technology, Division

of Itek Corporation

Sunnyvale, California 94086

## ABSTRACT

*General design tradeoffs for a 1553A interface* are reviewed and a set of design goals identified. A description of one design approach is given. The description covers signal level conversion, decoding, control, and subsystem to interface communications.

## BASIC CONSIDERATIONS

The design of a 1553A interface requires some functions that are common to every system. First, some sort of circuit is needed to receive the high level differential signals from the bus and transform them to normal system levels. Circuitry is also required to perform the reverse function, driving signals from the system to the bus. These two functions are basically analog. A method of retrieving the timing information from the incoming signal is needed. This function can be accomplished in either an analog or a digital fashion. Decoder and encoder are needed to perform the conversion of Manchester serial data to parallel TTL NRZ data and the reverse.

At this point, host system requirements vary. The two remaining interface functions are control and data exchange with the host system. 1553A requires a system to examine incoming bus commands and take different actions dependent on the data therein. This function can be built into the interface, or handled by the host system directly (Fig. 1). The latter is attractive for saving design time and hardware, but is much more difficult than it appears. Assuming that the host system has a computer, the interface would normally provide an interrupt to indicate that data has been received and service is needed. The most common point for this to occur is after a remote terminal has received

PRECEDING PAGE BLANK

and decoded an incoming command word. It is probable that the decoder would not initiate the interrupt until the word is complete and parity has been checked. The host CPU now has a maximum of five microseconds to begin a response on the bus, if that is the correct action. First, the computer must reach a point at which it can accept interrupts. This in itself may take more than the time allowed, but let us assume that the interrupt is accepted and recognized in a negligible amount of time. The CPU must now store its operating context and retrieve its interface context. Again, this can be an extensive procedure, but we will assume only a minimal time loss. Now the CPU must fetch the word from the interface and evaluate the address field to see if the command pertains to this system. If the answer is yes, the command word must be examined further to evaluate subaddress, mode, and receive/transmit fields. Now the CPU knows what action to take and can begin a response. Some further time is required to transform the response initiate instruction from the CPU into a waveform properly coded on the bus. To accomplish all of this in five microseconds or less, as measured at the bus, is extremely difficult for most operating systems. As the process must be followed through address identification for every command word on the bus, the host system CPU must be able to spend an appreciable amount of time in interface servicing.

In many cases these two requirements, that of response time and that of operating time, are enough to rule out direct interface control by the host system. If the control function is built into the interface, a second problem arises. The interface can service the bus and accept and store incoming data, but it must also make this data available to its host system. The choice of data transfer method should be tailored to the host system's needs. I/O or DMA modes may be acceptable if they do not require excessive CPU time.

## GOALS

With these problems in mind, some goals were set for the serial interface unit, or SIU. First, the interface should be versatile enough to be readily adapted to other systems. The design effort was estimated to be significant and a redesign task for every new 1553A application should be avoided. This required the interface to be self-contained, requiring only power, clocks and data interface from its host system. This in turn implies an independent interface able to handle completely the protocol of the bus. The space allotted for the interface was one 6 x 7 card, equivalent to 105 16 pin I.C. spaces. It was required that the SIU contain two totally independent bus interfaces, and that each operated under software control by the host system.

## THE RECEIVER

Meeting these goals proved to be an interesting design task. The description of the design starts with the most rigidly defined point, the bus itself. Two identical bus connections, consisting of the specified isolation resistors and the transformer are on the card. Each channel transformer drives a hybridized analog transceiver circuit. The hybrid filters out high frequency noise on the input, amplifies the waveform, and applies it to two comparators. One comparator is set to fire when the differential input voltage exceeds a positive threshold level. The other fires when the waveform is more negative than a

negative threshold level. Both thresholds can be adjusted externally to the hybrid. When a comparator is active, its output is a TTL logic one. These two logic outputs of the hybrid, one denoting a positive differential voltage on the bus, the other a negative bus voltage (Fig. 2) are passed on to a digital sampling circuit.

Both the timing and data information in the bus waveform are contained in the direction and time between zero crossings, but due to noise and other problems, these points may be difficult to detect. It may be noted, however, that the time between the waveform exceeding a positive threshold and exceeding the following negative threshold equals the time between zero crossing for symetrical waveforms (Fig. 2). Therefore, the rising edges of the two hybrid outputs provide a shifted but real zero crossing indication. These outputs are then sampled by a 16 MHZ clock, looking first for one rising edge, then the other. Each detected rising edge loads the output of a 16 MHZ pulse width counter into a FIFO and clears and restarts the counter. The polarity of the measured pulse is also loaded into the FIFO. This sampling approach avoids the task of reconstructing the bus timing and provides the first step in synchronizing the bus waveform to system timing. The FIFO accomplishes the second step of synchronization by providing for a separate input and output clock. It also provides a "rubber band" data storage, allowing the output circuit to ignore received pulses for a few cycles without losing data. Each channel has its own pulse measurement circuit and FIFO.

## THE DECODER

The decoder function of the SIU is implemented in the form of a state machine. A flowchart can be drawn for the decisions required to decode the stream of pulse width and polarity information coming from the bus through the digital receiver. Each decision is based on all previous decisions within the current word and the current pulse width data. The flowchart can be directly converted to hardware in the form of a state machine (Fig. 3). As each digital receiver will record pulses at a maximum rate of less than two MHZ, a state machine running at a 4 MHZ rate can handle both receivers. Keeping track of the two channels merely requires two state latches alternately active. On the SIU, a two MHZ clock provides channel control. When the clock is low, channel zero is being decoded; when the clock is high, channel one is being decoded. The state machine must keep track of its place within an incoming word, so it automatically provides a bit count function. It also easily accepts the job of tracking and checking parity. The state machine provides serial NRZ data as an output, and flags to indicate its status to a control processor. The data is accumulated in a buffer file register. The buffer file is an eight bit by 16 word two port RAM, where one port is serial and is dedicated to the decoder and the other is parallel and dedicated to the control processor. The decoder status flag outputs are written into a status flag register on command of the decoder. The write to the status flag register is used as a channel identified decoder interrupt to the control processor. The status flag register contents are read by the control processor. Interrupts are sent at the end of each byte of the word being decoded. This allows the processor to operate in a look ahead mode, relieving the more critical timing cycles. A control flag register is loaded by the control processor with control flag information identified by channel. The decoder state machine can select to read these flags instead of FIFO outputs. When the decoder selects the flags

as inputs, one bit of the serial output of the buffer file data is also available.

During the transmit function, when the SIU is driving data onto the bus, the data to be transmitted is taken from the buffer file by the decoder and sent in TTL Manchester format directly to the transceiver hybrid. Transmit control signals are also provided by the decoder to the hybrid. The hybrid performs the functions of level translation, amplification, and waveform smoothing and drives the bus transformer.

## THE CONTROL PROCESSOR

The Control Processor performs the message control function for the SIU. An eight BIT architecture was chosen as the optimum space to speed tradeoff. A 16 BIT wide path would have taken too much space and a 4 BIT wide path could not have run fast enough to service both channels for two simultaneously arriving 16 BIT words. The Processor uses 2901's as ALU and register storage with 16 additional registers provided externally. A two bus architecture is used to allow the 8 BIT processor to execute a 16 BIT memory interface operation in one cycle (Fig. 4). The 2901's and the buffer file input data from the upper bus and output data to the lower bus. The external registers input data from the lower bus and output to the upper bus. The memory interface registers, the address register, data in register, and data out register, load or drive upper BYTE data from or to the upper bus and lower BYTE data from or to the lower bus. This allows the Processor to load the upper BYTE of a memory reference word from the external registers and the lower BYTE from the 2901 registers to the memory interface registers in one micro instruction. The upper bus also can drive data to the microcode address registers or be driven by the emit field of microcode. The first allows the Processor to directly use memory data as a microcode address and the second allows data such as masks to be stored in microcode.

The microcode address is generated from one of three sources (Fig. 5). A microsequencer provides both a microprogram counter and a branch register. The branch register is loaded either from the microcode data field or the upper bus. Unconditional branch control is from microcode and conditional branching uses the zero and negative flags from the 2901's. The third microcode address source is the interrupt branch register, or IBR. The register drives the bus only when interrupts are being polled. It is an 8 BIT four word RAM with the same data inputs as the microsequencer. The IBR is addressed by the highest priority interrupt present. The contents of the register are loaded at the end of an interrupt service with the microcode address to be executed first when that same interrupt is serviced next. With all processor registers dedicated, the interrupt branch system allows the Processor to step from one interrupt service routine directly into any other interrupt service routine in only one cycle. The reason for this interrupt driven organization is the unpredictability of the processor's sequential operations. Two interrupts are from the decoder, one for each channel. The third interrupt is from the host CPU, and the fourth is available to the host system for general use. The four interrupts are prioritized in two groups. Group one contains the interrupts from the decoder and has the highest priority. Within the group, the priority is not rigid. Channel zero is given nominal precedence, but with both channel zero and one interrupts pending, the channel currently being serviced will assume a lower priority. Group two interrupts are from the host system

and have the same floating priority within the group.

## FIRMWARE

The interrupt driven processor operation combined with the timing limitations make the microcode task difficult. The decoder operates over a word, sending two interrupts to the Processor for each word on each channel. The Processor operates on a message basis, but must service only a byte of a message at a time. The microcode instruction essentially covers the full message length, but must be driven from one sub-instruction, micro-routine, to the next by decoder interrupts. During a given period, all interrupt levels may have instructions in progress. The Processor will perform one micro-routine in a given interrupt level, and then step to the next micro-routine for the highest level interrupt present. The interrupt prioritization guarantees that the Processor will accept for service an interrupt on any level before the next interrupt on that level will occur. If each channel receives a command word at the same time, both will send decoder interrupts to the Control Processor at the end of the first BYTE received. The Processor must accept the highest priority interrupt, process the flags and data from the decoder, make any decisions necessary, and load control flags for the decoder. It must then do the same for the second channel. As the decoder will examine the control flags for each channel and send two new channel interrupts to the Processor at the end of the second BYTE received on each channel, the Processor must be able to service both of the original interrupts in no more than ten microseconds, or each in less than five. At a 250 ns cycle time, this allows only twenty or less micro instructions to service a BYTE of data. If a memory reference is required, no more than twelve micro instructions are allowed. These timing requirements highlight the importance of the rapid interrupt service context switching.

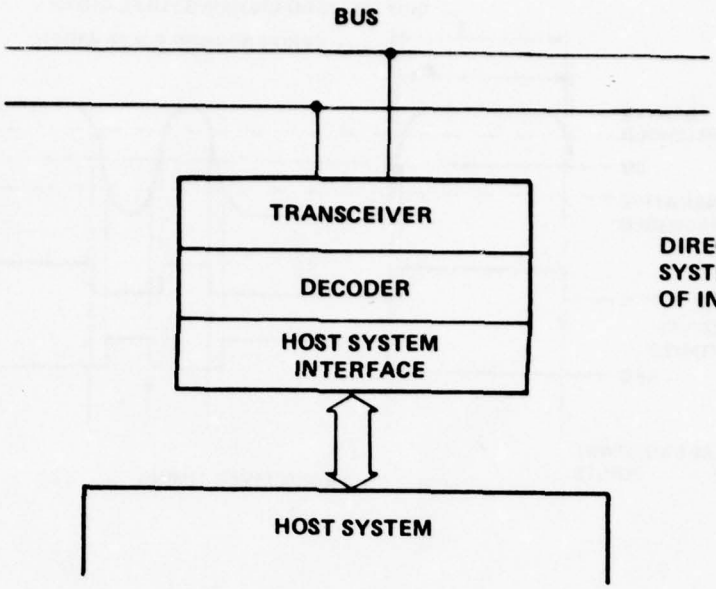
## SOFTWARE

The basic space limitation of the SIU makes it impractical to hold all interface operating parameters in the interface itself. Such data as sub-address data buffer pointers and mode command response words are kept in the host system main memory. The interface has its own port into one section of this memory, allowing immediate access to flag words, pointers, and multiple data buffers. One memory location is designated as an SIU control word. The host CPU writes an SIU control word to this location and then sends an interrupt to the SIU. When the SIU services this interrupt, it reads this memory location and then zeros it to inform the host CPU that the command has been acknowledged. On power up, the host CPU loads tables into memory that provide pointers for all active receive and transmit sub addresses and all mode commands. When the SIU receives a command word from the bus, the Control Processor uses the transmit/receive bit and the sub address field as a vector into this table. The content of the word accessed is the memory address of the correct transmit or receive data buffer. If the sub address field of the command word is all zeros, the Control Processor uses the word count field as the vector into the mode command table. As these vector tables are in the host CPU's main memory, it can change the data in the tables at any time. This allows the CPU to accumulate data for transmission in a working buffer, and then simply enter that buffer's address in the vector table to make the data available to the SIU. For incoming data, the CPU puts the address of an empty

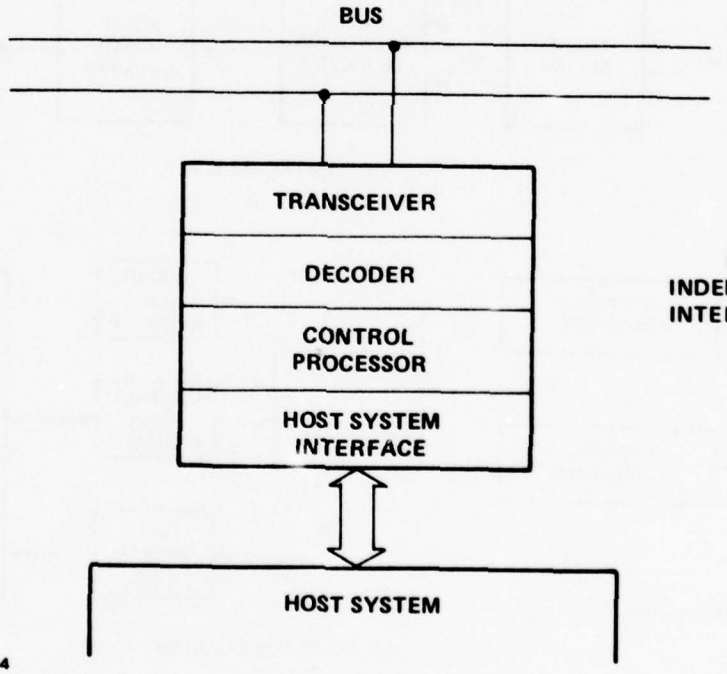
buffer in the vector table. When the CPU receives an interrupt from the SIU, it examines the location in memory that contains the SIU status word. This directs the CPU to the buffer that is now full. The CPU can now change the vector table pointer to another empty buffer and process the received data without moving it and with no danger of it being overwritten. This approach to interface-to-CPU data handling requires a minimum of CPU time and programming.

## SUMMARY

The Serial Interface Unit contains two totally independent 1553A interfaces on one 6 x 7 card. Each interface may be microcoded to be either a remote terminal or a control terminal. They may operate separately or as a redundant pair. The algorithmic decoder is implemented in standard MSI latches and PROMS. These PROMS may be re-programmed to handle variations or changes in bus protocol. Its small size, standard components and flexibility make it an attractive alternative to LSI implementations. The interface control processor handles the complete bus interface function. The host CPU merely receives notification of a completed message from a remote terminal. For a control terminal, the host CPU sends one interrupt to initiate a message and receives an interrupt upon completion of the message. Host system communications with the interface are through main memory, avoiding more time consuming I/O operations. As no data shifting is needed, interface support software requires only small amounts of code and CPU time. It is possible for the control processor to be programmed to perform auxiliary tasks in support of the host CPU. It may also be programmed to be a stand-alone unit. The SIU requires only basic inputs from a host system and is therefore easily appended to any existing or future system.



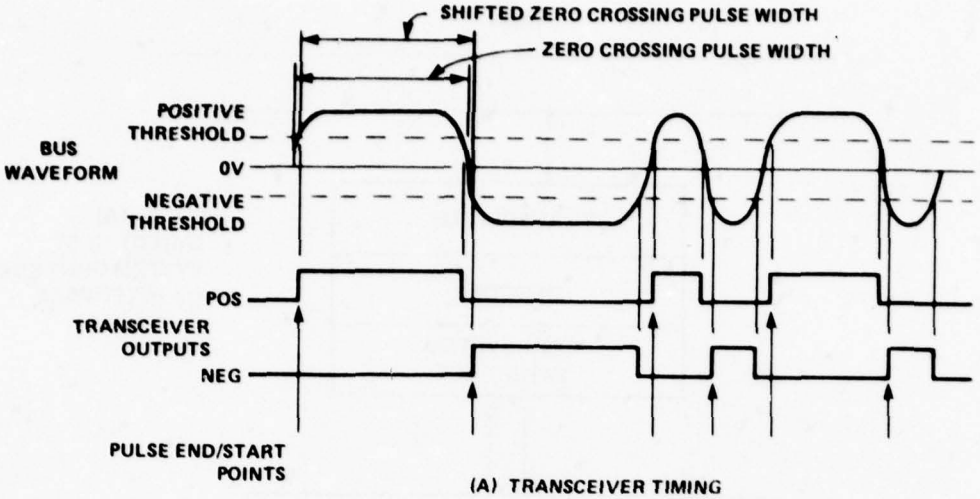

(A)
DIRECT HOST SYSTEM CONTROL OF INTERFACE



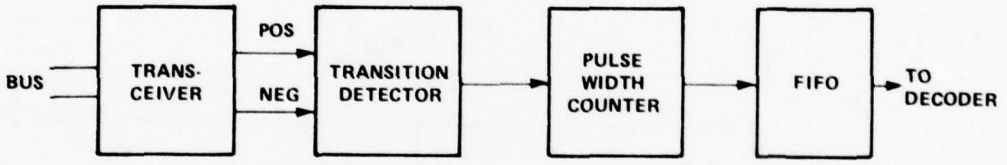

(B)
INDEPENDENT INTERFACE

7810-4181-2094

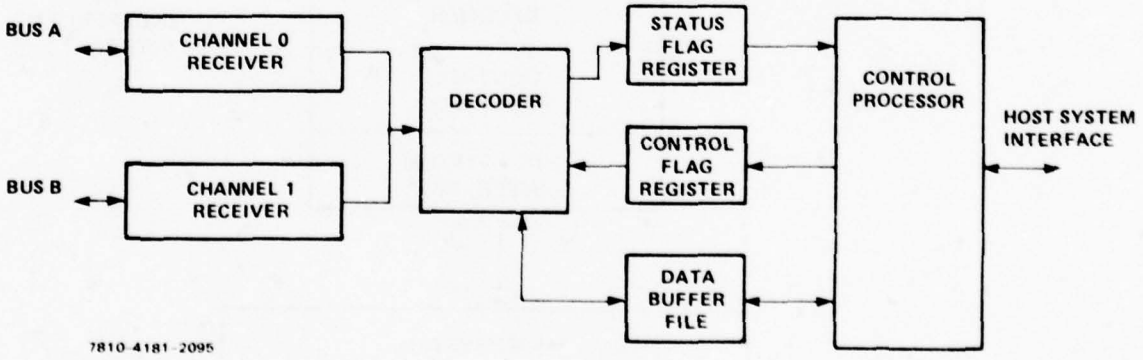
FIG. 1




(A) TRANSCEIVER TIMING

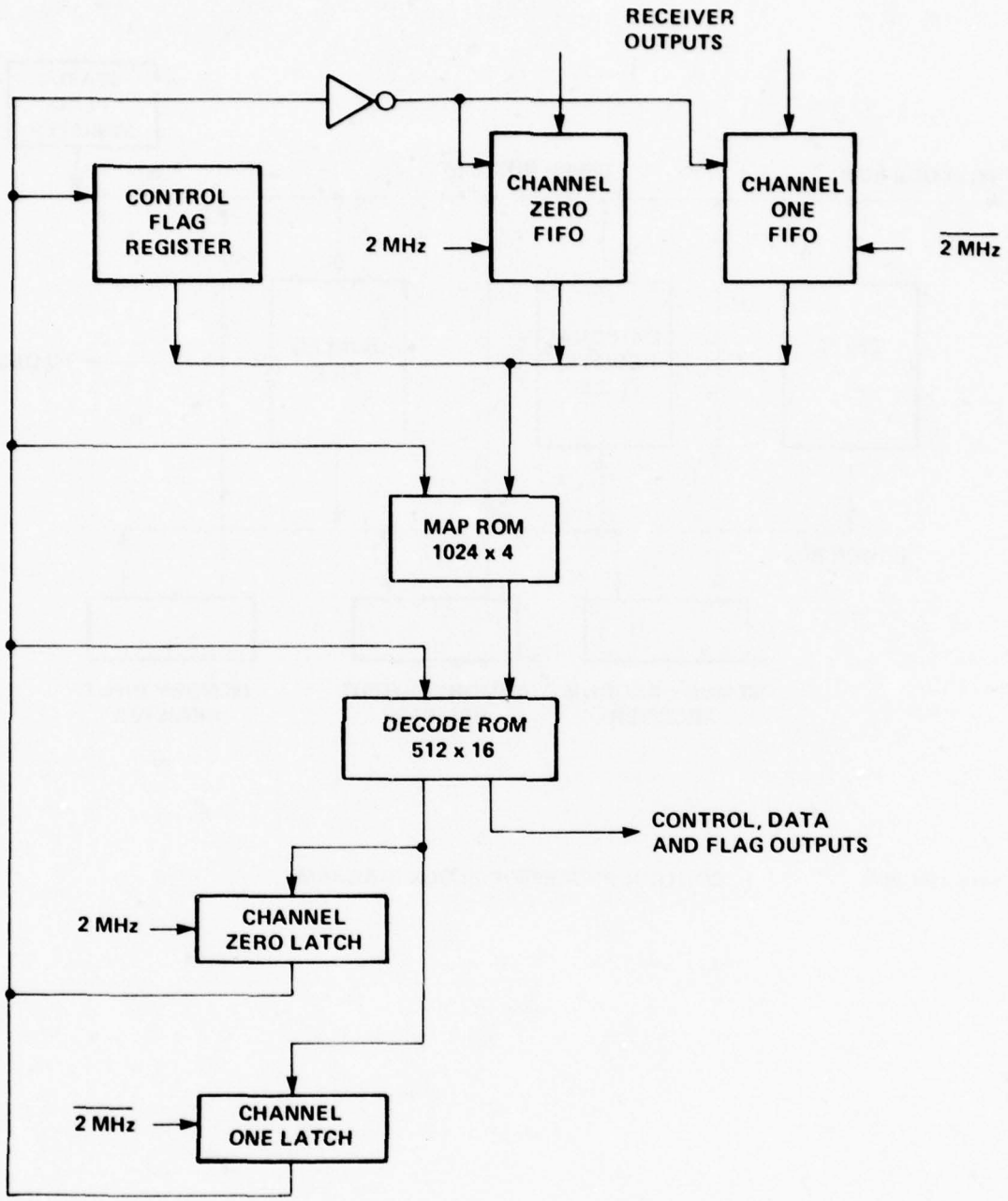



(B) DIGITAL RECEIVER




(C) SIU BLOCK DIAGRAM

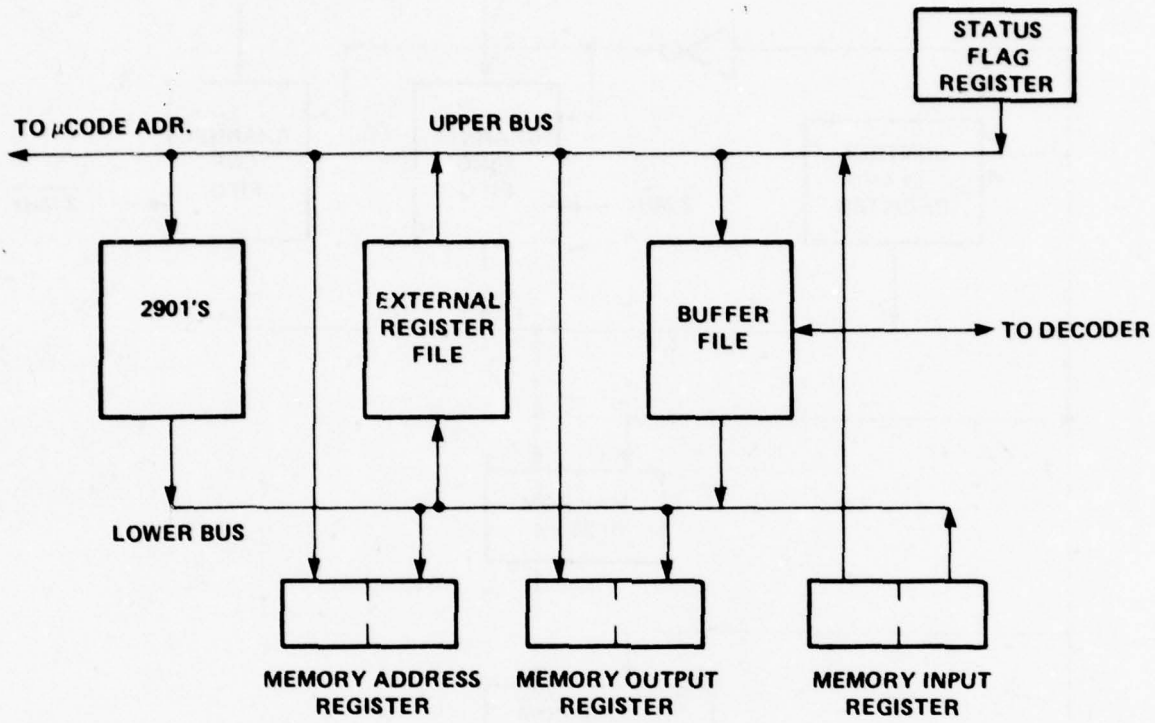
FIG 2.




7810-4181-2096

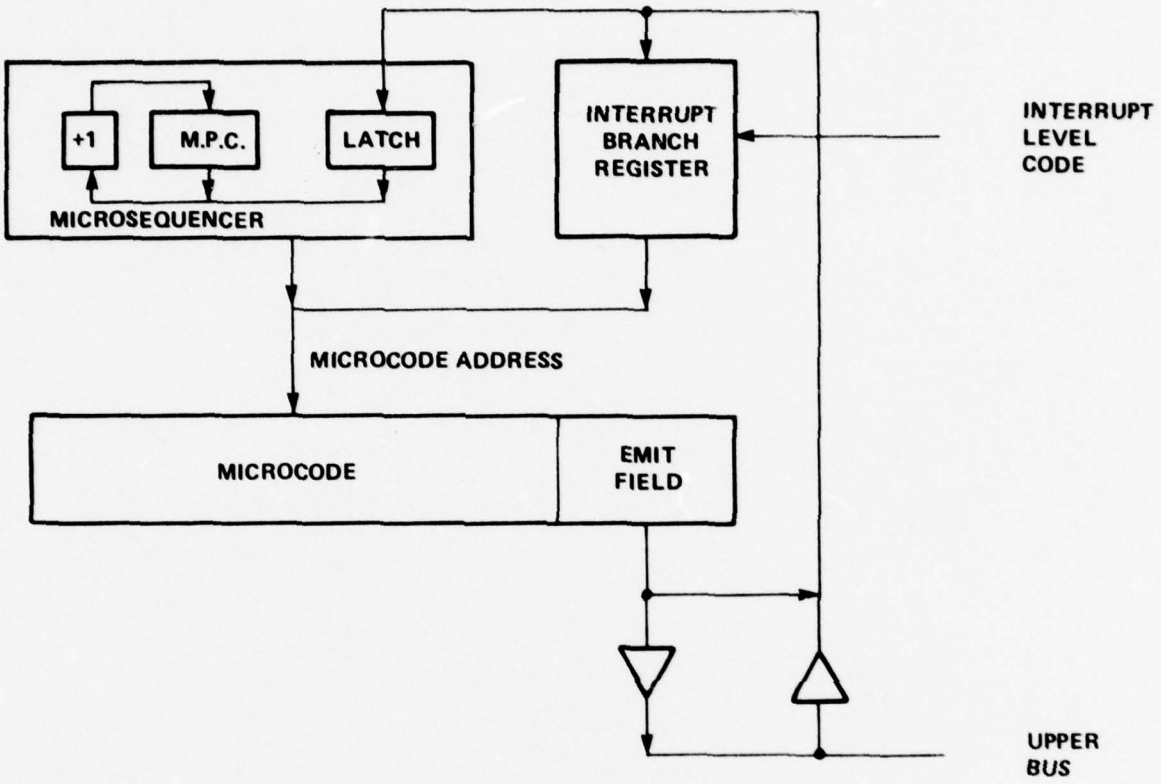
DECODER BLOCK DIAGRAM

FIG. 3.




7810-4181-2092 CONTROL PROCESSOR BLOCK DIAGRAM

FIG. 4.




7810-4181-2093

MICROCODE BRANCH BLOCK DIAGRAM

FIG. 5.

AFSC MULTIPLEX DATA BUS CONFERENCE

10 - 12 OCTOBER 1978

# A FLEXIBLE MULTIPLEX TERMINAL INTERFACE UNIT

Remigius Shatas and Jerry Blevins

SCI Systems, Inc.
8600 South Memorial Parkway
Huntsville, Alabama 35802



PRECEDING PAGE BLANK

ABSTRACT

This paper describes the architecture, features, and operation of one family in a spectrum of Flexible Multiplex Terminal Interfaces (FMTI) available at SCI. This family of Bus Control Units (BCU) is a set of multi-function bus interfaces, each tailored for operation with a specific host processor. The BCU design provides a data bus interface unit with direct memory access, programmable interrupts, built-in-test, and programmable operating mode.

A BCU can operate as a MIL-STD-1553A bus controller or remote terminal. Additional operating modes are bus monitor and bus analyzer, as defined elsewhere in this paper. Another available feature is simulation of all RT's on a given bus (up to 32). It is not implemented on all BCU's because of space constraints.

The BCU design employs a bipolar bit-slice microprocessor for superior flexibility. The standard operating modes occupy approximately 800 words of microcode, while the architecture allows utilization of up to 4096 words. All operational features are firmware controllable. As a result, the upgrade to MIL-STD-1553B can be accomplished by a straight-forward microcode update. Firmware commonality among BCU family members is greater than 90%, with common source code for all bus processing modules. This results in more stable microcode, since a "bug" fixed in any one application will be automatically applied to all BCU's.

## INTRODUCTION

The continuing evolution of digital processor technology as applicable to aircraft data bus systems has stimulated significant advances in the flexibility and capability of the data bus interface unit. The continued improvement of component and circuit technologies has permitted the data bus hardware designer to achieve these results, and the added benefits of higher reliability, lower cost, and improved adaptability.

A "family tree" of SCI's FMTI products is presented in Figure 1. It shows that the BCU is a natural continuation of development activities in multiplex technology dating back to 1966. A comparison of detailed function, size, and power characteristics of the FMTI products is given in Figure 2.

FMTI EVOLUTION AT SCI

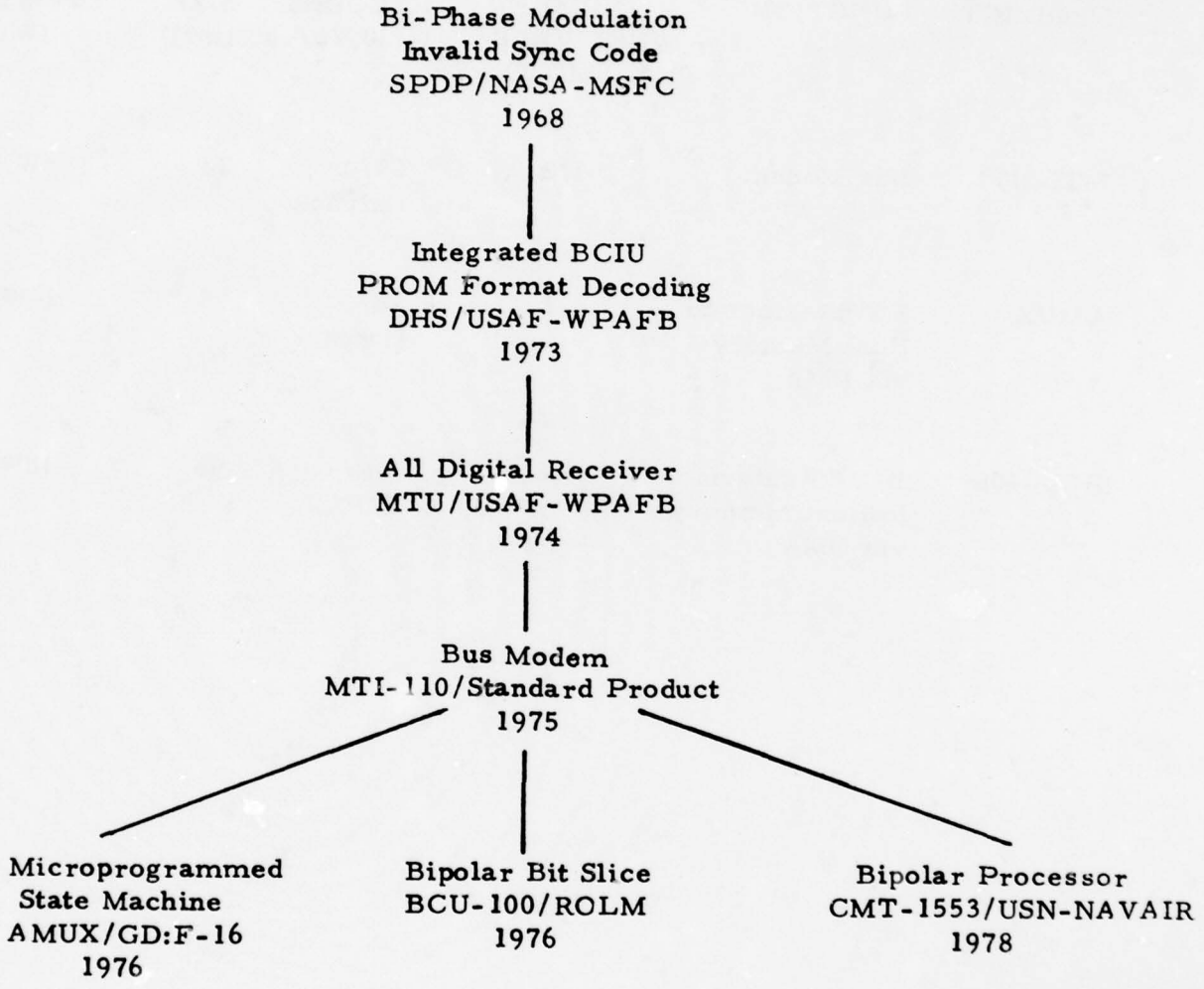



Figure 1

## FMTI SPECIFICATIONS

| PRODUCT | FUNCTION | NUMBER RECEIVERS/ TRANSMITTERS | IC COUNT 14, 16, 20/40 | SIZE ($in^2$) | POWER (W) |
|---|---|---|---|---|---|
| MTI-110 | Bus Modem | 1/2 | 63/0 flatpack | 24 | 5W |
| AMUX | RT Interface to Host Memory via DMA | 1/1 | 78/0 flatpack | 75 | 14W |
| BCU-400 | BC/RT Interface to Host Memory via DMA | 1/2 | 138/5 DIP | 96 | 18W |

Figure 2

## FUNCTIONAL DESCRIPTION

A BCU interfaces a host processor to dual MIL-STD-1553A data busses. It has two basic functions: communicating with the host processor via programmed I/O (PIO) and interrupts, and transferring data to or from the 1553 bus via direct memory access (DMA). Normally, these functions are mutually exclusive - the BCU cannot be initialized or interrogated while engaged in bus activity. However, the BCU status register can be read and a shutdown request issued, at any time by the host processor. Also, device flags can be tested, if provided in the host's architecture.

Before it can commence data transfers, the BCU must be initialized. This involves setting the contents of at least two internal registers - operation mode and activity pointer - by PIO commands. Other registers may be required in certain operating modes.

After initialization, the interface operates by interpreting the contents of a series of channel control blocks (CCB) obtained from the host memory via DMA. Each CCB, containing interface directives, MIL-STD-1553A text, and a link to the successor block in the chain, provides sufficient information to enable the BCU to operate without further host intervention. The BCU then executes the transfers specified by successive CCB's until the chain is exhausted or an error is detected in a bus transmission. In either case, a maskable interrupt is generated by the BCU, and bus data transfer ceases.

Both the host device code and the 1553 RT address can be specified by field-alterable straps. This is useful when multiple BCU's are installed in a single host. A default operating mode can be selected on the board, to facilitate operation of multiple units on the same bus with common software. All switch settings are accessible to the driver software.

When operated in the MIL-STD-1553A command response mode, the BCU can act either as a bus controller (BC), in which it initiates all information transfers, or as one or more remote terminals (RT), which can exchange data with the bus controller or another RT. If the multiple RT option is included, however, a given BCU cannot function as both the sending and receiving RT in a terminal-to-terminal transfer.

## ARCHITECTURE

The BCU hardware is organized into three primary elements: a serial digital interface with dual MIL-STD-1553A data busses, an internal 4 MHz 16 bit microprocessor and associated logic, and a parallel interface with the host processor. Figure 3 is a block diagram of the BCU showing the major functional blocks.

BCU BLOCK DIAGRAM

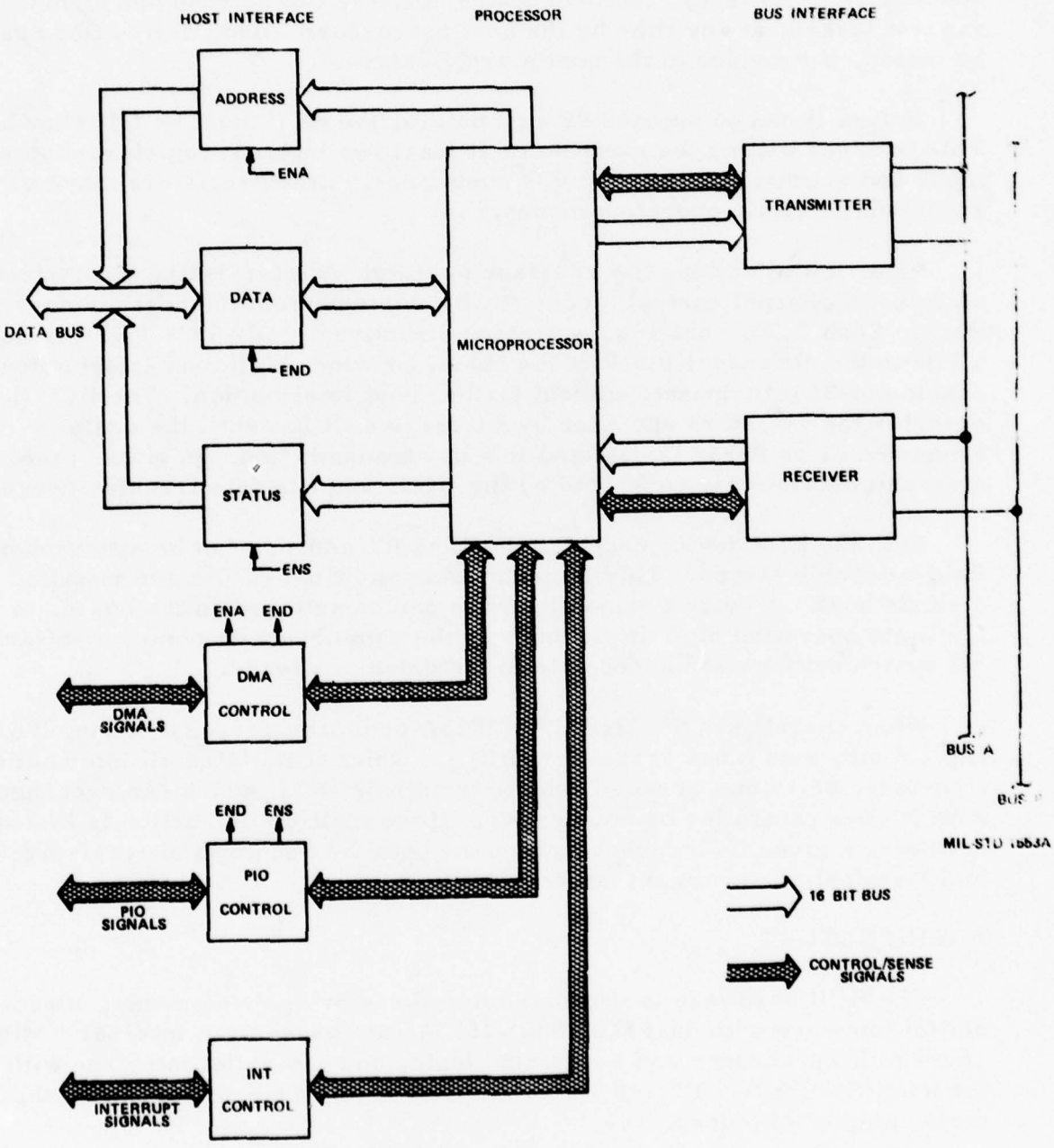



FIGURE 3

## Host Interface

The host interface contains three full-width registers which are shared among DMA, PIO and INT functions. This allows DMA operations to use the address and data registers and PIO reads to access the status register while the BCU is executing CCB's. Obviously, the DMA, PIO, and INT control logic must be tailored to the particular application, but this involves less than 20% of the BCU hardware design.

DMA control is performed by independent hardware on all BCU's because the timing requirements exceed the response time of the $\mu$P. Control information consists of DMA read/DMA write commands and a busy flag. BCU operation is largely unaffected by changes in DMA latency or transfer rate. Long latency times (19 $\mu$s max) can be tolerated in the BC mode at the cost of longer intermessage gaps. Latency times for successful execution of the RT mode can be no greater than 11 $\mu$s. If a DMA request is not honored in time, the firmware will request an interrupt after setting the appropriate bit in the status register.

PIO control is performed in firmware wherever feasible to reduce parts count. Obviously, the status input and shutdown commands are decoded in hardware to permit $\mu$P independence. The PIO software interface is elegant on asynchronous machines, such as the PDP/LSI-11 family. In synchronous environments (NOVA, V77) the software interface is non-standard, because BCU status must he tested before issuing a new PIO request to avoid possible overrun conditions. Many architectures include a series of device control and sense flags which can be directly utilized for this purpose.

Interrupt control is also achieved by dedicated hardware because of response time constraints.

## Processor

The $\mu$P employs bipolar bit slices (2901) to achieve 4 MHz instruction cycle times. It features a pipeline register at the PROM output, full carry look-ahead, 16 bit shift/rotate operations, and a 32 word mask PROM to allow single bits or selected fields to be isolated in a single microcycle. Microcode size is 1K x 40 bits, with expansion possible to 4K words, since a 12 bit instruction address space is standard.

Two classes of micro-instructions redefine the 40 bit instruction word. ALU instructions use 39 bits to specify operation code, register selects, data input source, output destination, mask selection, and shift type. They pass control only to the next sequential micro-instruction. Sequencer instructions use between 28 and 32 bits to control next address generation.

The full complement of 2910/2911 address control orders is supported, including two and three-way conditional branches, subroutine resting to four levels, conditional looping to a count of 1023, and vector branches.

An assembler and support software package was constructed to aid in firmware development. The assembler features an easily comprehensible input format; the resulting code can be followed without consulting detailed flowcharts. The firmware is organized into functional blocks: PIO handlers, diagnostic mode, BC mode, RT mode, and support subroutines are all maintained as separate modules. These are assembled separately and the relocatable object code is linked to form a load module and PROM programming tape. A sample of the source code is given in Figure 4.

1553 Bus Interface

The serial interface provides all functions specified by MIL-STD-1553A including signal characteristics, bus loading, and data validation. The transmitter, under processor control, provides a MIL-STD-1553A output signal with specified amplitude, rise/fall time, output noise, and transmission rate into a twisted shielded pair data bus with up to 32 remote terminals. The design provides for short stub or long stub operation as a strap option and provides transformer isolation to the bus. Two versions of the bus interface are available. The standard design, which can be radiation-hardened, consists of two receivers and a single decoder constructed from TTL MSI. A later design incorporates two HD 15530 CMOS encoder/decoder LSI circuits to achieve active standby operation, where a valid command word on one bus will abort a transaction on the second bus. Two transmitters are provided on all designs. Waveform shaping per MDC A3818 can be provided as an optional feature.

## BCU OPERATION

After the BCU has received PIO commands to preset the necessary internal registers, and received a start command from the host processor, it commences operation in the selected mode. One or more channel control blocks effect the actual data transfer operations of the BCU. Each CCB contains a 16 bit pointer its successor, which can be located anywhere in memory. The BCU continues processing CCB's until it recognizes an end-of-chain, or detects an error condition, or is halted by a shutdown command from the processor. Each CCB is composed of a two word header, containing a channel control word (CCW) and the link address. The remainder of the CCB depends on the operating mode and is defined in the following paragraphs.

## SAMPLE MICROCODE SOURCE

```
        JUMP NEXTS IFN DMA.ACT                                  ;
DMAERR: OR D A TO R, A = SSTATUS, B = BSTATUS, IN = MASKS, MASK = 13.   ; DMA TIMEOUT, INTERRUPT HOST
        OR D A TO S, IN = MASKS, MASK = 3                       ; SET CCB ERROR
        CALL SRZ
        CALL SRT
        OR D A TO R, A = SW, B = SW, IN = DATA                  ; SW SKEL INTO SW
        MOVE 0 TO NONE, OUT = DMA.LOC                           ; REQUEST .DATA
        ZERO A TO NONE, OUT = DMA.IN
        AND D A TO NONE, A = ACC, IN = MASKS, MASK = 5.         ; CHECK T/R BIT
        JUMP RECENT IF FR                                       ;
;
;       MT TRANSMIT CODE
;
        CALL SBU                                                ; TRANSMIT SW
        COM A TO NONE, A = .DATA                                ;
        CALL SBE IFN FR                                         ; OUTPUT DATA IF .DATA <> 177777
        ZERO A TO NONE, OUT = T.TIME                            ;
        COM A TO NONE, A = .DATA                                ;
        JUMP RTEOM IF FR                                        ; QUIT IF .DATA = 177777
;
;       UPDATE MESSAGE COUNTER IF ENABLED
;
MESSCT: AND D A TO NONE, A = CCW, IN = MASKS, MASK = 15.        ;
        JUMP RTEOM IF FR                                        ;
        MOVE A TO Q, A = RDDA                                   ; GET MESSCT
        JUMP SELF IF DMA.ACT
        CALL SRA                                                ;
        MOVE D TO R, B = ACC, IN = DATA                         ;
        JUMP RTEOM IF FR                                        ; MSGCT = 0, IGNORE
        DCR R TO R                                              ;
        MOVE A TO NONE, A = ACC, OUT = DMA.DATA                 ; UPDATE HOST MSGCT
        ZERO A TO NONE, OUT = DMA.OUT                           ;
        MOVE R TO R, B = ACC                                    ; REFRESH OUR MEMORY
        JUMP RTEOM IFN FR                                       ;
MSGCTB: OR D A TO R, A = SSTATUS, B = RSTATUS, IN = MASKS, MASK = 14.    ; MSGCT = 0 JUST NOW
        OR D A TO R, IN = MASKS, MASK = 3                       ; CCB ERROR
        CALL SRZ                                                ;UPDATE CCB SUB ADDRESS INFO...
        CALL SRT
```

FIGURE 4

The CCW format is given in Figure 5. The MSB of the CCW is the CCB skip bit, which when set, causes the BCU to ignore the current CCB and continue to the successor CCB. Bits 8-14 of the CCW control interrupt generation. The remaining bits specify error generation options, transmitter and receiver selection, and two bits of option code uniquely defined for each operational mode.

The second word in a CCB is always the link address. This gives the address of the next CCB in the chain. If the skip bit is set in the CCW, the BCU immediately chains to the successor CCB without executing the current one. Also, if the stop bit is set, the BCU requests an interrupt without executing the remainder of the CCB.

Off Mode

Off Mode operation is used for diagnostic purposes and does not result in information transfer on the data bus. While the BCU is in the off mode, all internal registers may be read and written, and switch settings may be read. In this mode, the I/O data transfer to all internal registers may be verified, and the BCU can be initialized for the other operating modes.

Bus Controller

In this operational mode the BCU acts as a MIL-STD-1553A bus controller (BC) with the capability to transmit or receive data from RT's or to initiate RT to RT transfers and optionally capture their data.

The BC CCB format is given by Figure 6. In addition to the control word and link, the CCB header contains a pointer to the message data area and the number of data words (DWC) to be transmitted or received. A zero in the word count field indicates that no words with data sync are to be transmitted or received. The retransmit count specifies the number of retries to be attempted in the case of a transmission error. If the message is successfully transferred before the count decrements to zero, an error interrupt will not be generated, even if enabled in the CCW.

The stored message consists of a header containing the actual command word to be transmitted, followed by a word reserved for the received status response. In the case of RT-RT transfers, two command words and status areas are necessary. Message data words may follow the header, or may appear anywhere in the host memory. Note that the data pointer in the CCB header must be the address of the first data word to be transmitted, or point to a buffer large enough to contain the received data words.

CHANNEL CONTROL WORD

Bit 15 (MSB) Skip this CCB if set.

Bit 8-14 Interrupt Generation. Generate a CPU interrupt as defined below:

- Bit 14 set: Interrupt CPU immediately.
- Bit 13 set: Interrupt CPU if the BCU status register indicates a transmission error.
- Bit 12 set: Interrupt CPU if the RT Message Error flag is set in the status word.
- Bit 11 set: Interrupt CPU if RT Terminal flag is set in the status word.
- Bit 10 set: Interrupt CPU if a bit is set in 3 MSB of RT Status Codes.
- Bit 9 set: Interrupt CPU if a bit is set in 3 middle bits of RT Status Codes.
- Bit 8 set: Interrupt CPU if a bit is set in 3 LSB of RT Status Codes.

Bit 4-7 Error Block Generation. Generate an invalid message as specified below:

- Bit 7 set: Bad sync in data, command and status words.
- Bit 6 set: Bad parity in data words.
- Bit 5 set: Bad parity in command/status words.
- Bit 4 set: Inhibit transmitter timeout (shutdown).

Bit 3 Transmitter Selection

- Bit 3 set: Transmit on secondary bus.
- Bit 3 clear: Transmit on primary bus.

Bit 2 Receiver Selection

- Bit 2 set: Receive on both busses.
- Bit 2 clear: Receive on transmit bus only.

Bits 0-1 Option Code.

Figure 5

BUS CONTROLLER
CCB FORMAT

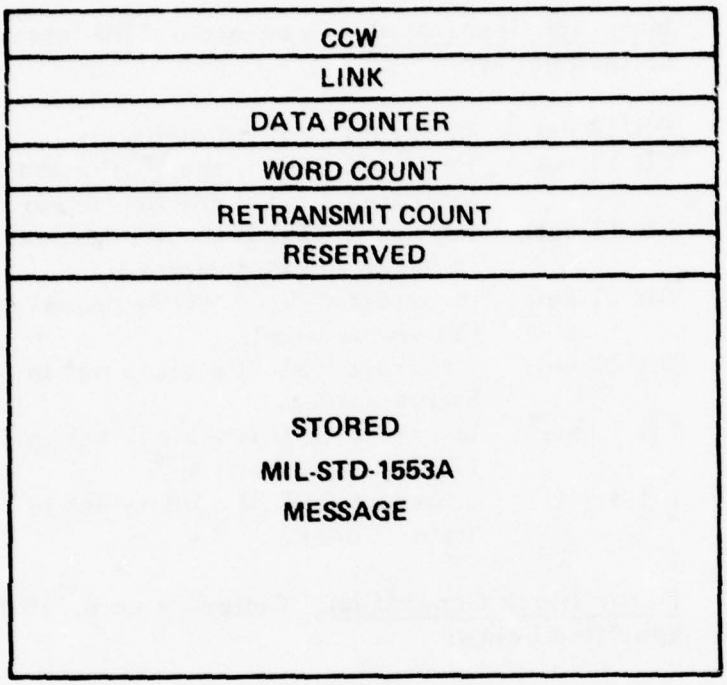



FIGURE 6

The function to be performed by the BCU in the BC mode is specified by the option code in the CCW. Codes are specified for BC to RT and RT to BC information transfer, as well as RT to RT with and without the copy option.

Remote Terminal

In the RT operation mode, the BCU acts as a MIL-STD-1553A remote terminal with Subaddress recognition capability. The BCU can either transmit data to a BC or receive data from the BC, depending on the state of the T/R bit in the command word it receives. An internal register is provided to specify the maximum time (in microseconds) that the data bus can be idle without generating an interrupt. This feature is enable in the CCW option field.

The RT CCB format is given in Figure 7. The CCB header contains the CCW, link, and status word skeleton. The skeleton is OR'ed with the parity error detect and RT bus address before being transmitted back to the BC. Thus except for BIT purposes the user should ensure that only the 10 LSB's are set in the skeleton word. Also, this skeleton is updated whenever any command word is detected on the bus; thus to change the transmitted RT status word, the user need only to store a new skeleton in the CCB.

The remainder of the CCB consists of 64 four-word entries, one for each Subaddress T/R pair. The first word of the entry is the address of the data area while the second is a message counter. If the data pointer for a particular combination is zero, a BC reference to this combination will generate an interrupt. If the pointer is all ones, the reference will be ignored by the BCU, causing a message error to be detected by the BC. If enabled in the CCW the message counter is decremented whenever a combination is referenced. When the counter becomes zero, an interrupt is generated. This is useful when a particular combination is to be accessed a specified number of times. The last two words of each entry are reserved for future expansion.

If the timeout counter is enabled, a dead-line detected for longer than the specified period will generate an interrupt. The message counter and timeout updates are controlled by the CCW option code.

The BCU can operate in full compliance with MIL-STD-1553A requirements for RT operation, including the optional mode control feature. The mode control can be accomplished by zeroing the data pointer for subaddress 0 to transmit and receive in the CCB. This will generate an interrupt whenever SA 0 is referenced by the BC. Measured response time to a BC command word is 4.8 $\mu$s.

REMOTE TERMINAL
CCB FORMAT

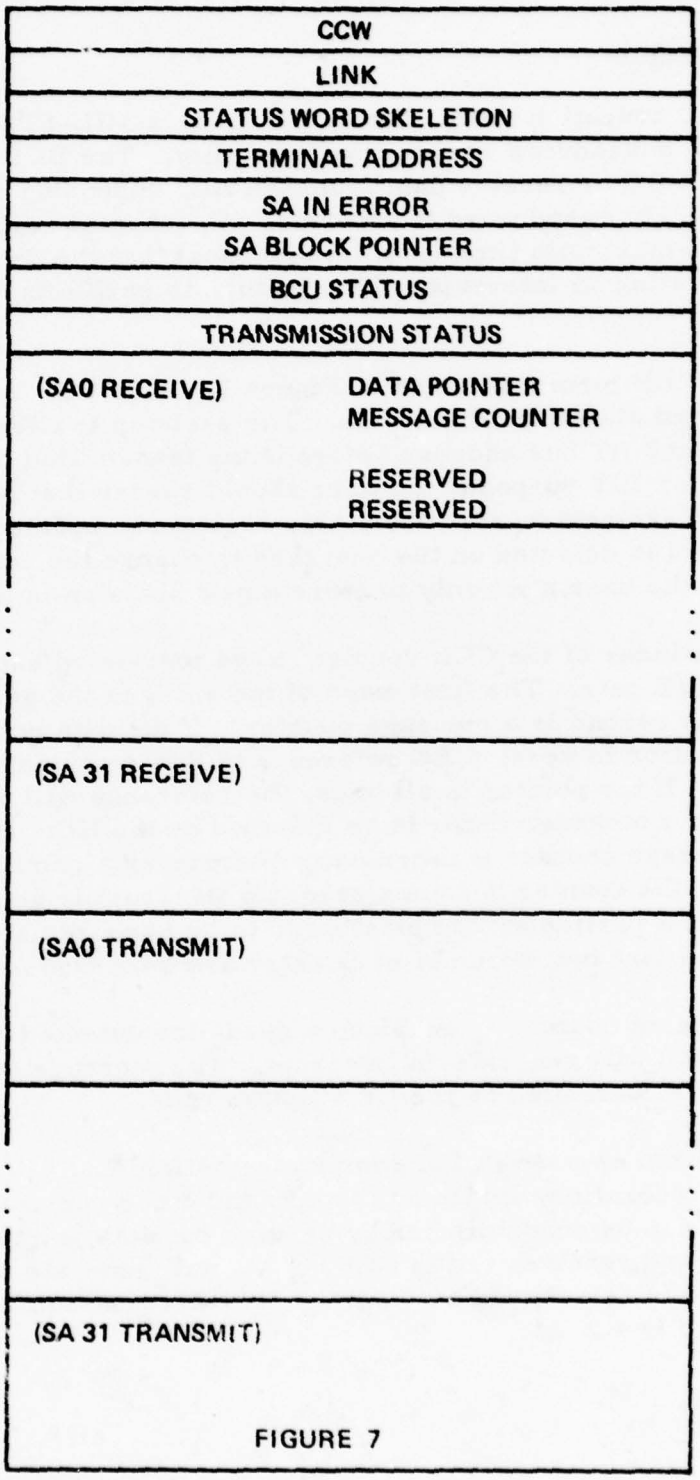



FIGURE 7

It should be noted that the BCU will remain in a particular RT CCB for an indefinite time; it will enter a new CCB only when commanded by the CPU, unless the skip or stop bits are set in the CCW.

Multiple Remote Terminal

This mode allows the BCU to respond to command words for up to 31 remote terminals, as long as it is not participating in both the transmit and receive functions of a terminal-to-terminal message. It is an expanded form of RT mode operation, and the same internal registers are provided. The CCB for the MRT mode consists of a CCW, a link word, and one or more RT CCB's. The format of the included CCB's is identical to that in the RT mode for standardization, but obviously the control words, and links are ignored.

If a bus error is detected, the BCU status and transmission status will be written into the CCB, as well as the terminal address and subaddress of the RT detecting the error, and the subaddress pointer. The CCB format is given in Figure 8.

Before initiating MRT operation, the number of terminals to be simulated must be set in an internal register. The BCU then determines the actual terminal addresses from the included CCB's.

Diagnostic

This operation mode permits the CPU to verify the BCU internal data paths at three levels: DMA, transmitter/receiver logic, and transmitter/receiver analog signals. Five tests are performed in this CCB, and the format is to read one or more words from the test input buffer, route them through the specified data path, and write them back to the test output buffer. The CPU can them compare corresponding pairs of input and output test words. In all tests, the output buffer size must be twice the input buffer size, since tests 2-5 also output the BCU transmission register, and test 1 adds the BCU status register. The size of each input buffer is specified in the CCB header, and can range from 0 to 255. Note that this allows the transmitter shutdown feature to be checked by tests 4 and 5.

The CCB format is given in Figure 9. Note that for test 1, the output buffer pair includes the DMA echo word and the BCU status register contents, while tests 2-5 write out the test result and the BCU transmission status register contents for every input word. A short description of each test follows.

Test 1: The μP reads a word from input buffer 1 and writes it to output buffer 1. This tests DMA input and output functions, as well as internal data paths. If a noncompare results, the

MULTIPLE REMOTE TERMINAL
CCB FORMAT

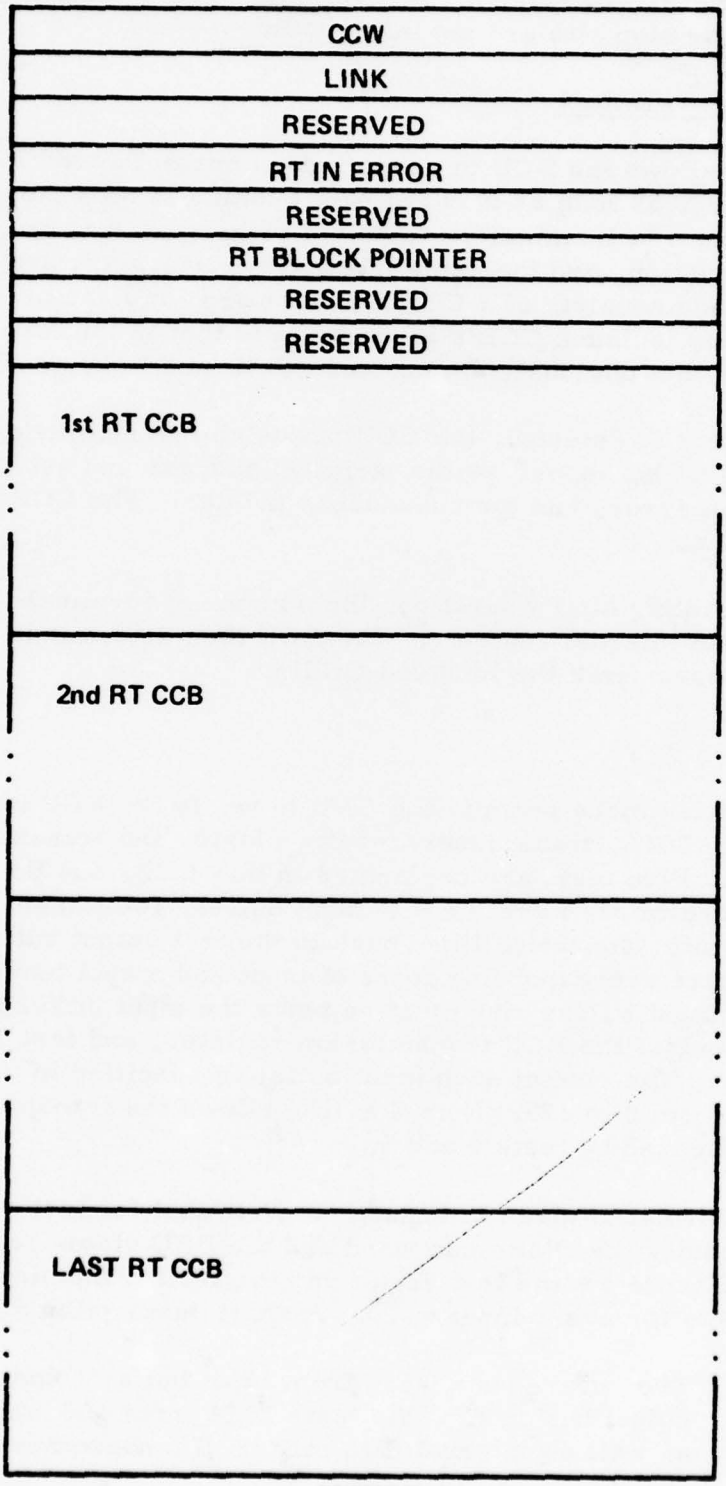



FIGURE 8

DIAGNOSTIC MODE
CCB FORMAT

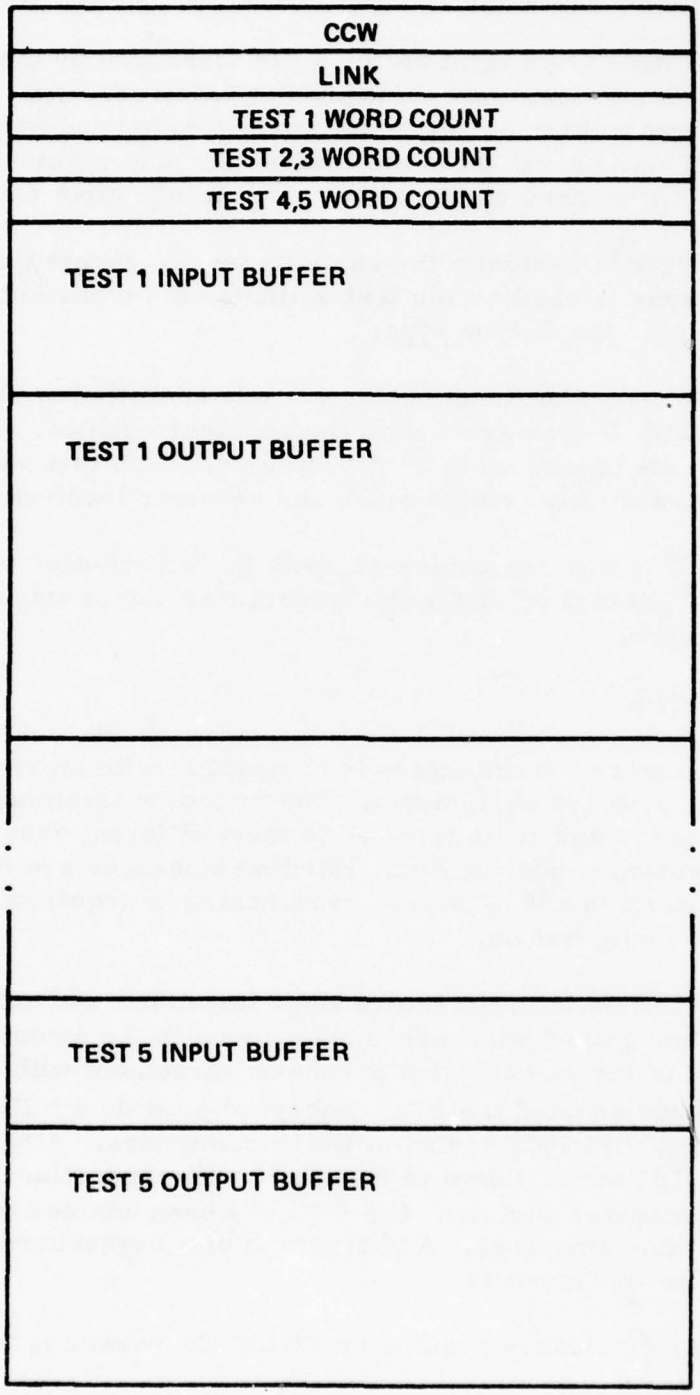



FIGURE 9

CPU can isolate the problem to either DMA input or DMA output registers via PIO tests if the $\mu$P is functioning.

Test 2: Words from input buffer 2 are formatted by the transmitter control logic and routed to the receiver, bypassing the transmitter output. The words are formed with data sync. This test validates the transmitter and receiver logic for D-type sync without appearing on the output data bus.

Test 3: This is basically the same as test 2, except that command sync is used. This test validates the transmitter and receiver logic for C-type sync.

Test 4: Words from input buffer 4 are transmitted on the data bus with D-type sync, with the receiver enabled. The received data is written to output buffer 4. This test validates the transmitter output stage and receiver front-end for D-type sync.

Test 5: This test resembles #4, with the substitution of C-type sync. This test validates the transmitter output stage for C-type sync.

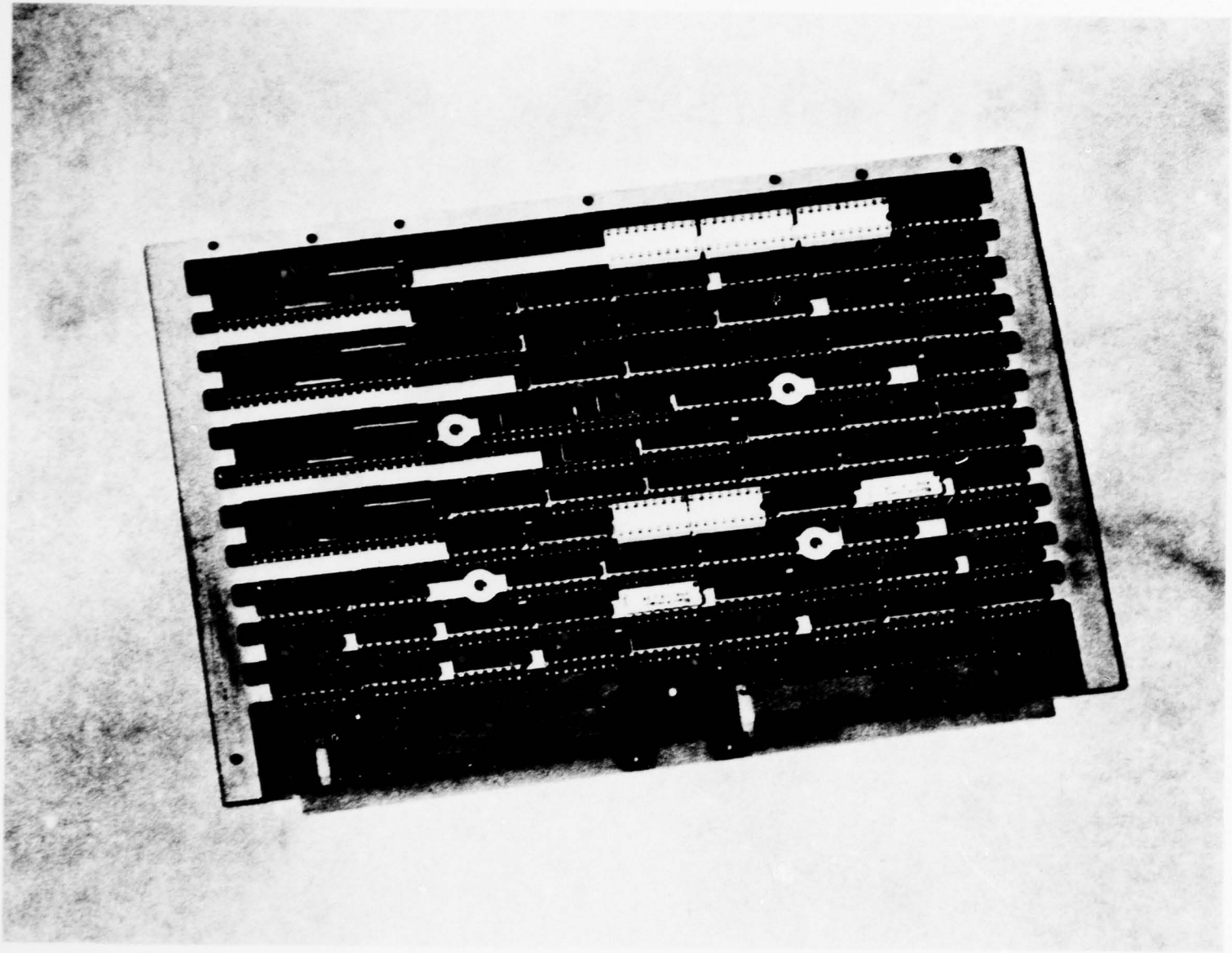
## BCU APPLICATIONS

The BCU design and architecture is compatible with most of todays MIL-STD-1553A avionics equipments. The micro-programmability features of the BCU enable the unit to be tailored to meet different user requirements primarily via firmware modification. Hardware changes are only required at the subsystem interface and of course repackaging is required to meet the user mechanical configuration.

Due to similarities in architecture and compatibility of subsystem interfaces. the BCU has gained widespread acceptance in the avionics processor community. SCI under an exclusive purchase agreement with ROLM Corporation [1] has adapted the BCU concept to provide a MIL-STD-1553A interface for the ROLM 1600 series military computers. A photograph of this two card ROLM set is shown in Figure 10. In consortium with Sperry Univac [2] Minicomputer Division, the BCU has been adapted to the Sperry V77-400 series minicomputers. A photograph of a brassboard unit delivered to Sperry is shown in Figure 11.

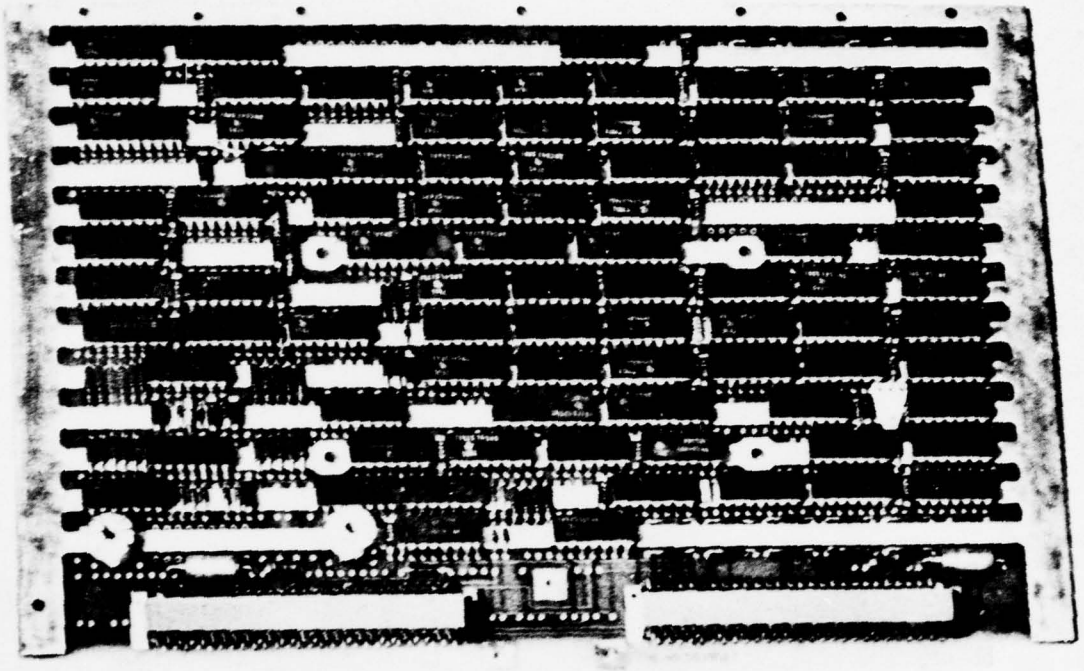
[1] Permission for disclosure granted by ROLM Corporation, Santa Clara, California.

[2] Permission for disclosure granted by Sperry Univac Minicomputer Division, Irving, California.



MIL-STD-1553A BUS CONTROL UNIT BCU-100

M1023



MIL-STD-1553A BUS CONTROL UNIT BCU-100

M1022

Additional applications of the BCU in various stages of development are illustrated in Table 1. Each of these applications employ the same design differing only in the host interface packaging configuration.

## ACKNOWLEDGMENTS

The authors wish to acknowledge James L. Voight, senior staff engineer at SCI who contributed substantially toward the design of these products and Donald H. Ellis, Applications Manager at SCI who contributed substantially in the conceptual development of the BCU and FMTI product lines at SCI.

## LIST OF ABBREVIATIONS

| | |
|---|---|
| BC | Bus Controller |
| BCU | Bus Control Unit |
| CCB | Channel Control Block |
| CCW | Channel Control Word |
| DMA | Direct Memory Access |
| DWC | Data Word Count |
| FMTI | Flexible Multiplex Terminal Interface |
| INT | Interrupts |
| MTI | Multiplex Terminal Interface |
| MRT | Multiple Remote Terminal |
| PIO | Programmed Input/Output |
| RT | Remote Terminal |
| SA | Subaddress |
| T/R | Transmit/Receive |
| $\mu$P | Microprocessor |

## BCU APPLICATIONS

| PRODUCT/ MODEL | HOST PROCESSOR | CURRENT DEVELOPMENT STATUS |
|---|---|---|
| BCU-100 | ROLM 1600 SERIES | PROTOTYPES DELIVERED |
| BCU-400 | SPERRY V77-400 | BRASSBOARDS DELIVERED |
| BCU-DG 16 | DATA GENERAL NOVA/ ECLIPSE | OPERATIONAL BRASSBOARD |
| BCU-LSI II | DEC LSI-II | BREADBOARD TEST |
| BCU-MDS 80 | INTEL MDS/8080,5,6 | FINAL DESIGN |
| BCU-NORD 50 | NORD 50 FAMILY | INITIAL DESIGN |
| BCU-HP 1000 | HP 1000 | INITIAL DESIGN |
| BCU-SEL 32 | SEL 32/75 | PLANNED |

TABLE 1

# THE DESIGN AND DEVELOPMENT OF AN LSI BUS INTERFACE UNIT

Andrew Gutierrez
Major Gary Pritchard

Air Force Avionics Laboratory
Wright-Patterson Air Force Base, Ohio

## ABSTRACT

The Air Force Avionics Laboratory (AFAL) is sponsoring an ongoing contractual program with Harris Corporation, GISD, Melbourne, Florida for the design and development, and demonstration of an LSI Chip-set Bus Interface Unit (BIU). The resulting design will comply with MIL-STD-1553B. Additional flexibility will be available due to the use of DAIS-like data registers and pointer tables used for host interfacing. The functional partitioning of the two chip-set is such that most Remote Terminal (RT) requirements can be fulfilled by the first chip (BIU #1) alone, while both chips (BIU #1 and BIU #2) can be used to implement the Bus Controller function. The chip-set is designed to replace nearly all of the logic between the multiplex bus analog to digital interface and the host (or host processor) back bus. The chip-set will be second sourced by Intersil.

This paper conveys, in a general sense, the design of the LSI BIU chip-set. A discussion of the problems and resulting trade-offs which must be made are presented. Not only were technology trade-offs made, but more importantly, bus system protocol trade-offs affected the BIU design. The very detailed technical details have been reserved for presentation in the accompanying paper entitled "An Adaptable LSI Chip-Set for Use on Multiplex Data Bus Systems," presented by Harris Corporation.

## BACKGROUND

The Air Force Avionics Laboratory (AFAL) is sponsoring the development of a large scale integrated (LSI) bus interface unit (BIU) for implementing MIL-STD-1553 multiplex data buses. The BIU development is part of an AFAL technology base development project which strives to exploit LSI technology to the advantage of military avionics systems by identifying common functions which have wide intersystem applicability and making these functions available as LSI circuits to system developers. The total BIU development program includes in-house development of a preliminary specification, contractual development of the BIU LSI circuits including production by the prime source and establishment of a second source, functional demonstration of BIU capability, and military qualification and nuclear radiation response characterization.

Harris Government Information Systems Division, Melbourne, Florida was awarded a contract in August 1977 for designing and developing the LSI BIU. To assure the BIU satisfies tri-service requirements, representatives from Army and Navy as well as from Air Force organizations have participated in periodic design reviews. The original design goal was a single LSI chip which could operate in several modes including operation in a stand alone remote terminal, a fixed sequence terminal, a smart terminal, or in a bus controller. Functionally the BIU was to be fully compatible with both MIL-STD-1553A and DAIS (Digital Avionics Information System) protocol and incorporate functions for future requirements not presently included in MIL-STD-1553A.

During the BIU development these goals were modified. It was found that all required functions could not be economically realized on a single LSI chip using Harris's silicon gate CMOS process. A cost-effective two-chip design evolved where chip #1 incorporates the functions required for a remote terminal and chip #2 incorporates the additional functions required for a bus controller. Also the BIU design schedule coincided with the period when the B revision to MIL-STD-1553 was being formulated. After careful review of the present and potential future users, MIL-STD-1553B was chosen as the baseline for the BIU design. Where appropriate the MIL-STD-1553B mode codes are implemented directly by the LSI BIU chips. The present BIU design is the result of several trade-off decisions. The complexity of the chips could have been decreased at the expense of requiring additional support circuitry for each use of the BIU chips.

More flexibility could have been incorporated by using external ROMs to define the protocol for each application, but this again would represent an additional cost to the system users. The present BIU design is compliant to MIL-STD-1553B, uses the DAIS defined stored program technique, data buffer structure, automatic (progammable) message retry capability and, in general, reflects the DAIS back-bus protocol. The low power Schottky TTL compatible BIU can serve as a DMA controller between the random access memory of the host computer in a smart computer, or be capable of performing the functions required for stand-alone operation in the case of a remote terminal.

## GENERAL DESCRIPTION

The LSI chip-set is designed to perform the interface functions between the multiplex bus analog receiver/transmitter unit and the host electronics. The analog receiver/transmitter (R/T) unit is the set of electronics which convert the manchester signals on the multiplex bus, be it fiber optic, twisted pair, or coax, to logically equivalent CMOS compatible signals which the BIU can process. Also, the R/T interface contains the mux bus line drivers which allow the BIU to transmit data. The driver circuits in the unit must be able to accept the bipolar zero and bipolar one output signals from the BIU and drive the appropriate mux bus line drivers. The R/T interface may contain filters to condition the mux bus signals, depending on the system design specifications. The R/T unit does not process the sync or data; this is accomplished by the BIU. The host electronics consists of the system or subsystem which needs communication via the mux bus. The host may or may not contain memory, although for most applications the host will consist of at least one processor and will contain memory which will be used for message buffering. Figure One depicts the mux bus/BIU/host interconnection.

The input to the BIU will accept a serial unipolar representation of the bi-phase data and, on the output, generates the logical equivalent of serial bi-phase data. At the host electronics interface, the BIU accepts instructions from the host and, by way of these instructions transfers parallel data to and from the host. The received data can be used immediately, or it can be stored, under DMA control, in the memory of the host for use at a later date. Status information is conveyed to and from the BIU by means of dedicated status lines and access to internal status registers.

Two major operating modes are available in the BIU: (1) In the Remote Terminal (RT) mode the BIU is responsive to the mux bus as the major driving source of BIU activity, and (2) The BIU in the Controller mode is responsive to both the requirements implied by the mux bus and the requirements supplied by the host. These two operational modes are reflected in the logical partitioning of the BIU chip-set. The first chip, BIU #1, can act as a stand-alone unit for use in RTs. For the additional capability required by the controller function, the second chip, BIU #2, is used to enhance the operation of BIU#1. Working together as a chip-set provides the full protocol, including mode codes, of MIL-STD-1553B and provides the register and data handling flexibility of a DAIS-like system.

The basic block diagram for BIU #1 is given in Figure 2. The manchester decoder provides synchronization with the incoming manchester data and feeds the rest of the chip-set separated data and clock. The manchester encoder adds the appropriate sync type and parity to the data and transmits the result to the mux bus line driver circuit. Both the manchester encoder and decoder are usable over a baseband rate of 1 to 5MHz. The Instruction Decode and Execution block is used to initialize Chip #1, and to drive it as required in the Bus Controller mode by Chip #2. The Instruction Encode and Execution Logic allows Chip #1 to convey the status on its internal operation, such as message received and message complete. The Error Detection Logic constantly monitors the integrity of the received messages. Conditions such as message too long, message too short, parity error, and manchester bi-phase errors are reported to the host. The Parallel Interface Logic can be configured to operate with either an 8 or 16 bit host system.

BIU #2 is block diagrammed in Figure 3. The basic design is a register stack with appropriate control logic. BIU #2 is used to enhance the operation of BIU #1. It can drive BIU #1 and the two chips thereby implement the Bus Controller function. When used with BIU #1, the input and output control lines of BIU #2 are split into two groups: those to and from the host, and those to and from BIU #1. The I/O data bus is shared between BIU #1, BIU #2, and the host. The basic interconnection structure is given in Figure 1.

Each of the BIU chips is packaged in a 40 pin dual in line, or a 40 pin hermetic chip carrier. The units require a single 5 volt power supply. The voltage level of the

power supply is constantly monitored by BIU #1 and if a glitch occurs BIU #1 will reset itself to a default "quiet" state. This reset condition is conveyed to the host and BIU #2 so that the decision whether to re-initialize the BIU can be made.

With both chips interconnected, the resulting BIU unit is responsive to the mode codes of MIL-STD-1553B. Although many of the mode codes are acted upon internal to the BIU, all of them are passed on to the host. Those which require further action, or those which the BIU cannot interpret, can be processed by the host. In addition, DMA indirect access and storage of data is accomplished through the use of pointer blocks, and message scenario control is accomplished by the use of instruction word pairs which are fetched and interpreted by the BIU Chip-set.

## SUMMARY

The design of the LSI BIU Chip-set provides hardware support for MIL-STD-1553B, although this was not the only goal of the program. The prime impetus was the design and development from two sources of a useful LSI Chip-set for the interfacing of avionics subsystems to multiplex data buses. Because of the wide support from industry and the military for MIL-STD-1553B, it was chosen as the baseline for the BIU design. Additional capability similar to that used in DAIS protocol was added for increased utility. Contributions added from the DAIS architecture were such things as indirect access of data via pointer tables, programmability through the use of two word instruction pairs, and the use of a shared input/output bus structure.

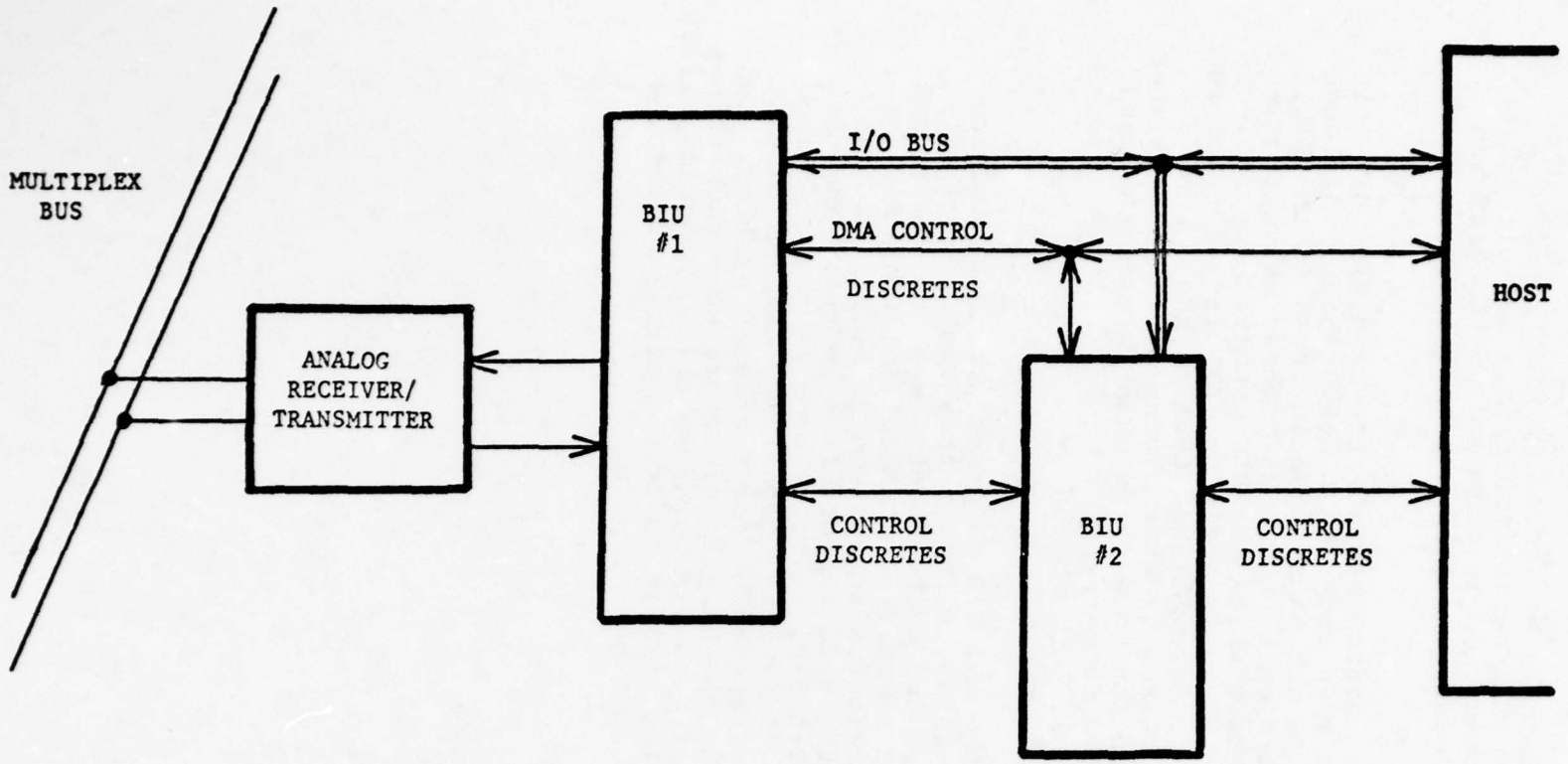
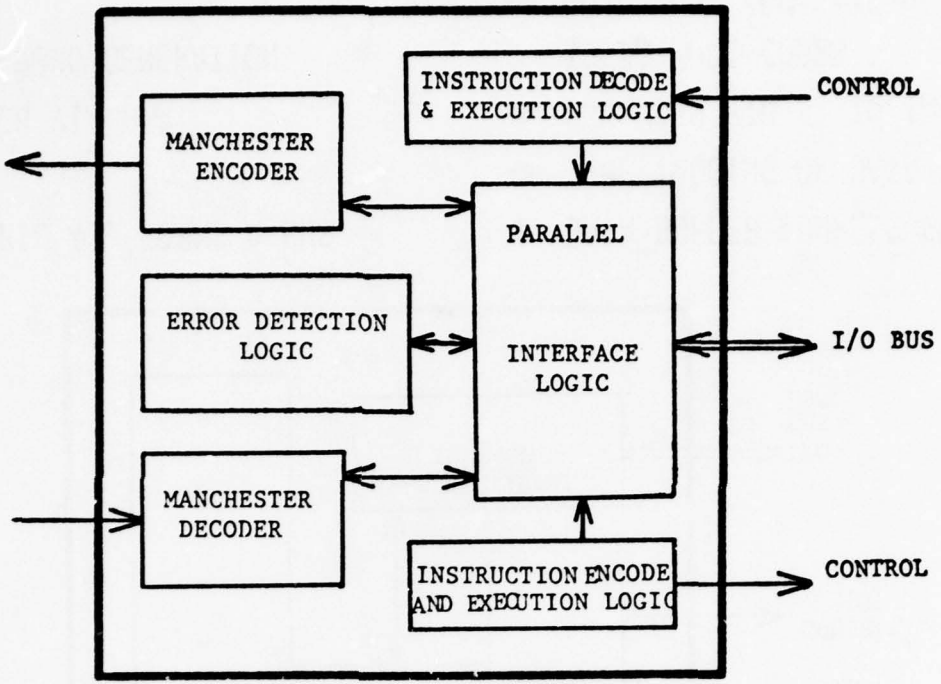



FIGURE 1. INTERCONNECTION OF BIU #1, BIU #2, AND HOST

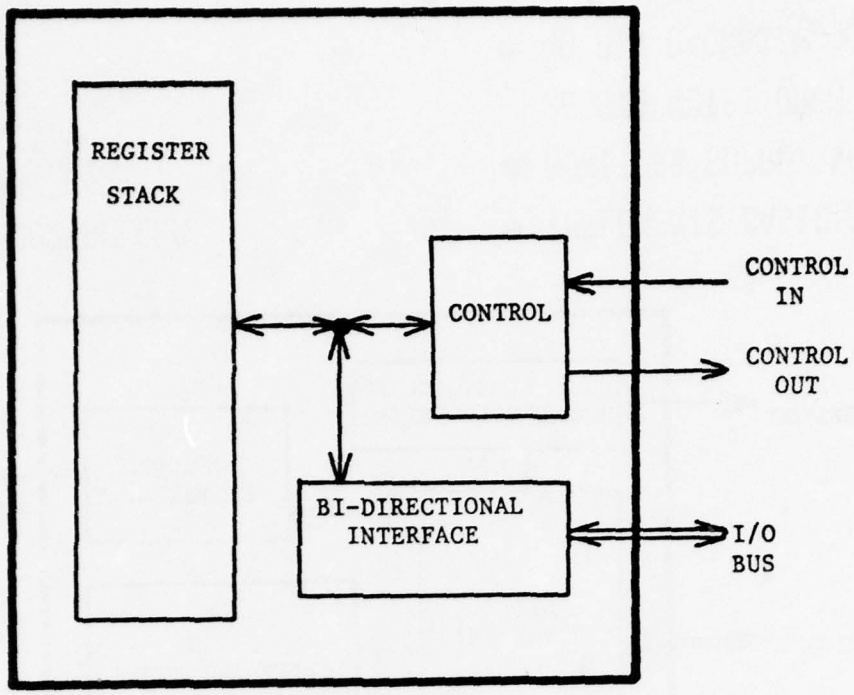
BIU #1




- STAND-ALONE REMOTE CONTROLLER
- 8 OR 16 BIT SYSTEM
- 1→5 MHz DATA RATE
- ERROR DETECTION
- IMPLEMENTS BASIC MODE CODE SET
- MONITORS SUPPLY VOLTAGE AND RESETS QUIET UPON GLITCH
- 40 PIN PACKAGE
- DMA CONTROL

FIGURE 2. BIU #1 BASIC BLOCK DIAGRAM

BIU #2




- WHEN USED WITH BIU #1, FORMS A BUS CONTROLLER
- INSTRUCTION FETCH VIA DMA
- MESSAGE-COMMAND-WORD GENERATION
- DATA BUFFER ADDRESS GENERATION
- TIME TAGGING OF DATA IN DATA BUFFERS
- ENHANCES BIU #1 TO IMPLEMENT ALL 1553B MODE CODES
- INTERRUPT GENERATION

FIGURE 3. LSI BUS INTERFACE CHIP NUMBER 2

# THE BUS INTERFACE UNIT
# AN ADAPTABLE LSI CHIP-SET

Daniel G. Yelverton

Harris Corporation,
Government Information Systems Division

## 1.0 Bus Interface Unit (BIU)

1.1 The BIU is an LSI approach to the implementation of the interface between the host electronics and the 1553B manchester data bus. A two chip approach was used due to the complexity of the system. One chip, BIU #1, can act as a stand-alone unit for use in a less complex system such as a remote terminal (RT). Where an additional capability is needed, the second chip, BIU #2, is used to enhance the operational capability of BIU #1. The following discussion will describe the use of 1) BIU #1 as a stand-alone unit, and 2) BIU #1 with BIU #2 as a chip-set.

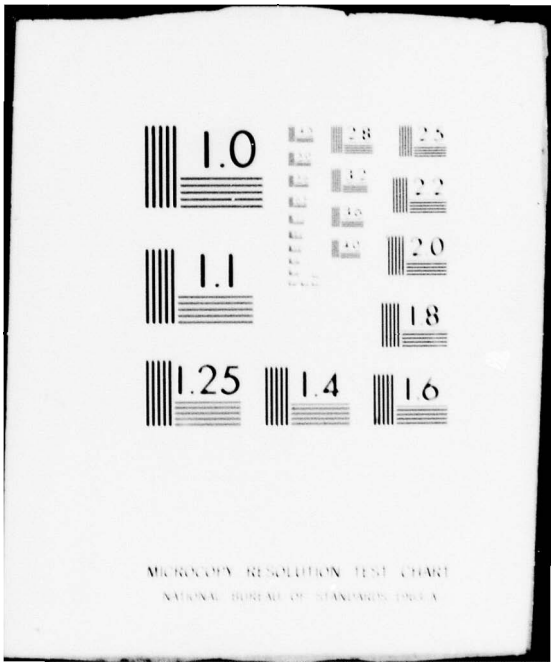
1.2 BIU #1 has many unique features which extend its system applicability. For instance, baseband data rate of 1 to 5 MHz; internal error checking; internal Busy mode; can almost completely handle activity between the host and the 1553 data bus. Figure 1.2-1 shows the basic block diagram of BIU #1 and Figure 1.2-2 shows the functional pin outs of BIU #1.

1.2.1 Initialization of BIU #1 to operate in the Controller mode is accomplished by enabling the control word onto the I/O bus and strobing the "Code Execute Strobe" input. The BIU is now in the operational controller mode. To transmit, a command word is enabled onto the I/O bus and the "Load CMD word" input is strobed. BIU #1 decodes the type of command loaded and executes it as such; i.e., Transmit, Receive, Mode (with or without data word), Broadcast, RT-RT. If data transfers to or from the host are required, then BIU #1 will activate the appropriate DMA control lines to

AD-A068 310 AERONAUTICAL SYSTEMS DIV WRIGHT-PATTERSON AFB OH F/G 17/2
PROCEEDINGS OF THE AFSC MULTIPLEX DATA BUS CONFERENCE (2ND), HE--E
OCT 78 E C GANGL
UNCLASSIFIED ASD-TR-78-34-VOL-2 NL

7 OF 9
AD A068310




1.0
1.1
1.25
1.4
1.6
2.0
1.8
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

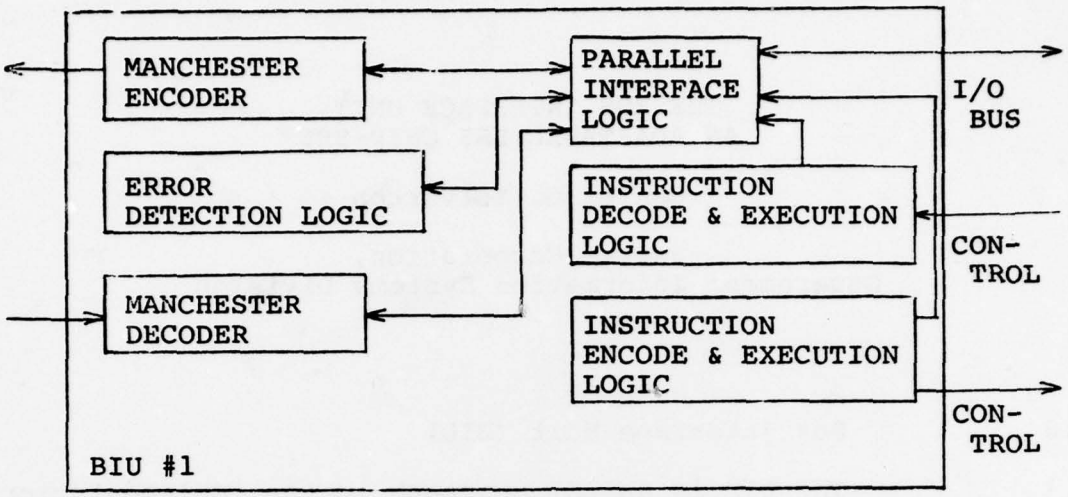



FIGURE 1.2-1
BIU #1 BASIC BLOCK DIAGRAM

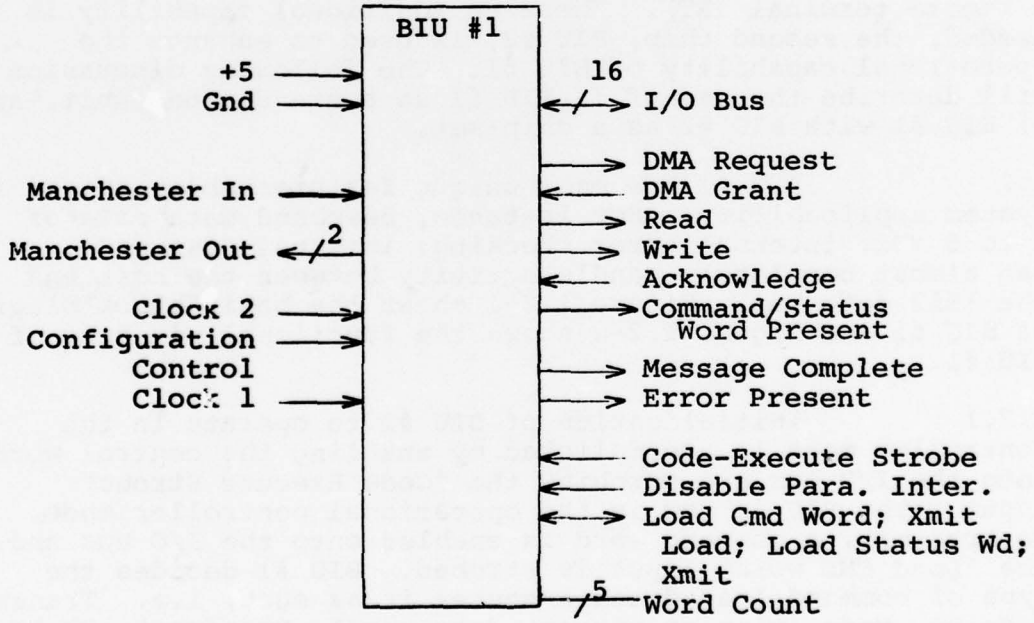
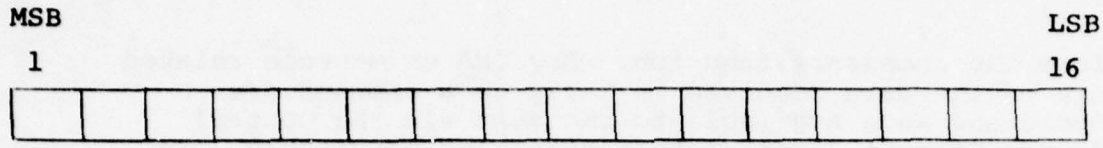
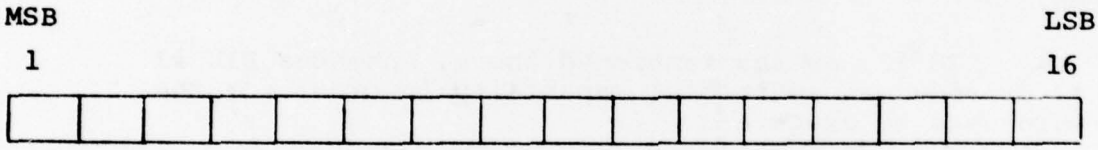



FIGURE 1.2-2
BIU #1 FUNCTIONAL PINOUT




| BITS 14 | 15 | 16 | | |
|---|---|---|---|---|
| 0 | 0 | 0 | ABORT | USED BY BIU #1 |
| 0 | 1 | 0 | RT RT | |
| 1 | 0 | 0 | ERROR REGISTER ENABLE | |
| X | X | 1 | REDEFINED AS BELOW | |




| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| E | D | C | B | A | MASTER/REMOTE | GO | BUSY | ON LINE SELECT | 0 | 0 | 0 | 0 | INHIBIT DMA ERROR | DYNAMIC BUS CONTROL ACCEPT | 1 |
| BOX ADDRESS | | | | | | | | | | | | | | | |

FIGURE 1.2-3

CONTROL WORD FORMAT

perform the necessary function. Any DMA or Message related errors on the data received in reply to a command are recorded and made available to the host via the Control Word/Code Execute Strobe method discussed previously. An indication of an error condition is given by the discrete output "Error Present". A message in process may be aborted by the same control word/code execute strobe method above.

1.2.2 Initialization of BIU #1 to operate in the remote mode is accomplished as in Paragraph 1.2.1. Once in this configuration, BIU #1 will answer any command addressed to it with either a normal mode command/response or a busy status depending on the logic state of the "Busy/Disable" input. In the normal mode, BIU #1 as a remote will decode all commands addressed to it. The commands decoded, data transfers, and error checking is performed by BIU #1 as described in Paragraph 1.2.1.

1.2.3 In an overview, BIU #1 may be used as a controller or remote and will perform all the DMA and error checking function related to the transfers being executed when used as a stand-alone device. Figures 1.2.3-1 and 1.2.3-2 show two simplified connections of BIU #1 in a stand-alone application.

1.3 BIU #2, as was mentioned above, enhances BIU #1. BIU #2 provides an additional capability in order for the two chip sets to execute:

1) instruction fetch and decode,
2) data buffer address generation,
3) message data handling,
4) tag word generation,
5) mode command execution, and
6) interrupt generation.

The basic block diagram of BIU #2 is shown in Figure 1.3-1. A functional pin out diagram of BIU #2 is shown in Figure 1.3-2.

1.3.1 Five major items are covered in this section:

1) data buffer structure
2) fetch cycle
3) command word generation
4) data buffer address generation, and
5) time tag word generation.

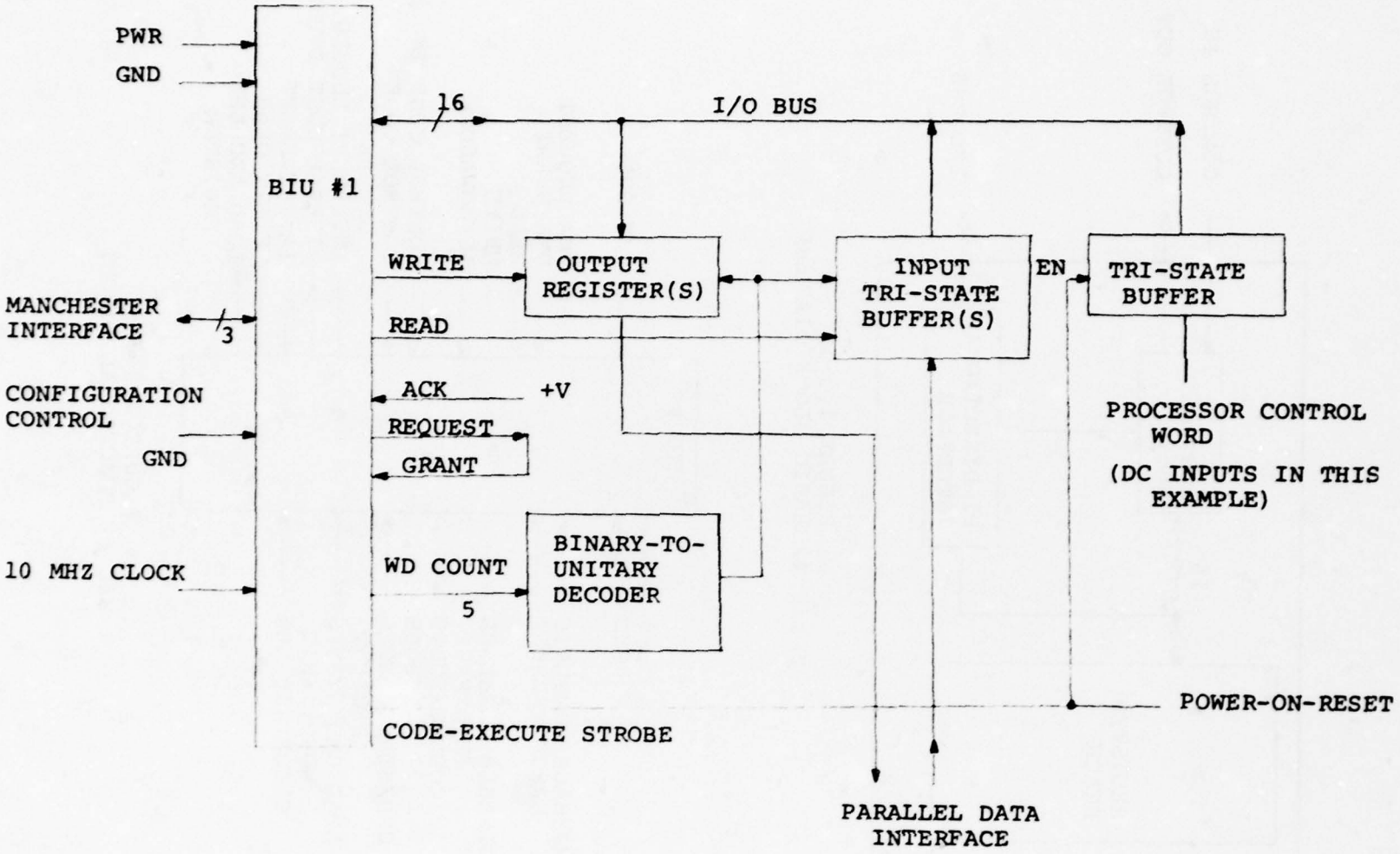



FIGURE 1.2.3-1

SIMPLE REMOTE TERMINAL

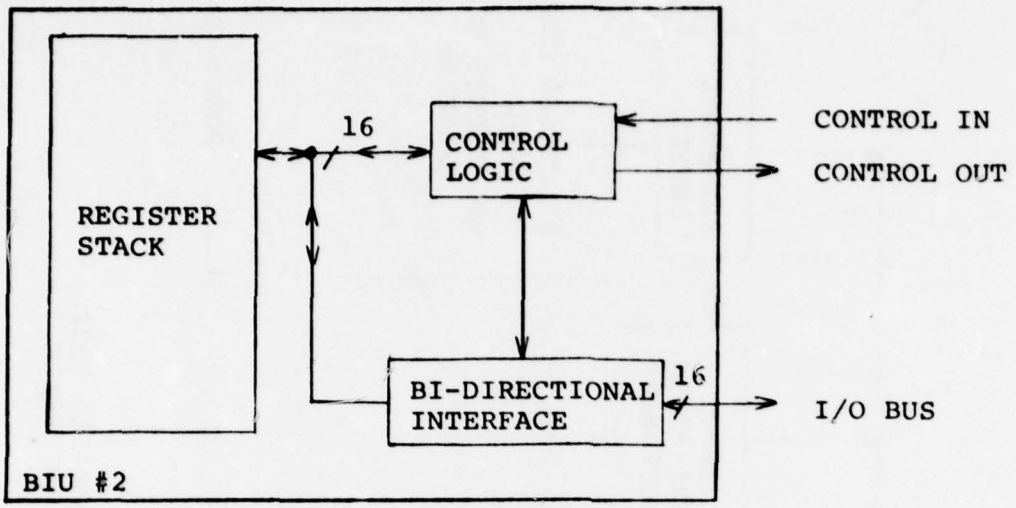



FIGURE 1.3-1
BIU #2 BASIC BLOCK DIAGRAM

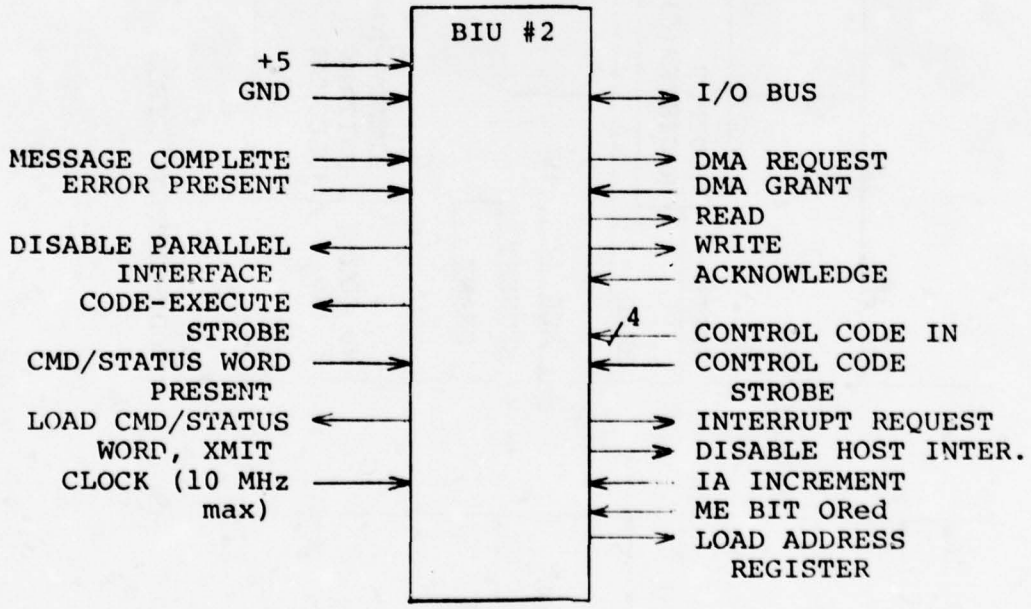



FIGURE 1.3-2
BIU #2 FUNCTIONAL PINOUT

The BIU assumes all message data is packed in data buffers accessed indirectly through pointer blocks. Each buffer contains N+1 addresses so a data buffer can hold a tag word in the first location with the N message words following. At initialization the BIU is given data for its Instruction Address Register (IAR) which points the BIU to its stored program instructions. Instructions there are arranged in pairs; the format is given in Figure 1.3.1-1. The BIU is also given a Base Address Register (BAR) word; and a Processor Control Word (PCR). In a fetch sequence the IAR contents are placed on the I/O Bus and loaded into an address register. The first instruction word is acquired via DMA. The incremented values of the IAR is again loaded into the address register and the second instruction word is acquired via DMA. (The IAR is incremented again internally to prepare for the next fetch).

Once the two instruction words are acquired, BIU #2 can construct the command words. Referring to Figure 1.3.1-1 BIU #2 compares its terminal address (available from the control word (PCR word) given to it when setup by the host) with the device address in the instruction words. If BIU #2's address is the same as the Receive-Device Address, then the command to be generated is a transmit command to an RT. If BIU #2's address is the same as the Transmit-Device Address, then the command to be generated is a receive command to an RT. If BIU #2's address does not compare with either device address, then an RT-to-RT pair of commands is to be generated. As part of BIU, BIU #2 checks to assure the Receive-Device Address is different from the Transmit-Device Address. (An interrupt condition is set when the two are the same).

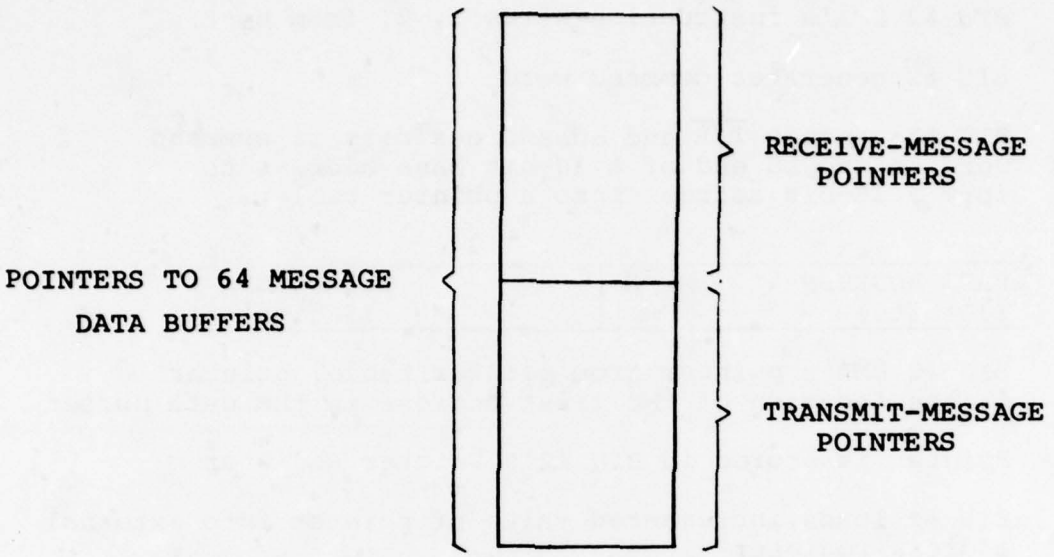
When an RT-to-RT set of command words is formed, no data buffer address is generated since the controller does not transfer data in that case. When an RT-Receive of RT-Transmit command is generated, BIU #2 generates a corresponding data buffer address. As mentioned above, BIU #2 is given a Base Address Register (BAR) word. BIU #2 appends 6 bits (the $\overline{T/R}$ bit and subaddress bits) of the command word to the LS end of the BAR word to form an address into a pointer table. The pointer, acquired from the table, points to the first address in the data buffer. This first address is reserved and the pointer to it is stored in the Pointer Register of BIU #2. The incremented value (pointing to the second address of the data buffer) is then loaded into the external address register--ready for use by BIU #1 when it executes its DMA transfers. Once the data buffer address is set up, BIU #2 is ready to transmit. From that point,

| 1 2 3 4 | 5 | 6 | 7 8 9 10 11 | 12 13 14 15 16 (LSB) |
|---|---|---|---|---|
| OP CODE RE-TRY | B | 1 | RECEIVE DEVICE ADDRESS | RECEIVE SUBADDRESS/MODE |
| WORD COUNT/ MODE CODE | | B | TRANSMIT DEVICE ADDRESS | TRANSMIT SUBADDRESS/MODE |

| IWI | BIT | DESIGNATION |
|---|---|---|
| | 1-2 | Normal OP Codes<br>Bit 1-2<br>00 = Halt BIU<br>01 = Link (Use seocnd word as address of next 2-word instruciton)<br>10 = No Operation (go to next 2-word inst.)<br>11 = Bus Operation |
| | 3-4 | Indicates number of retries (0, 1, 2, or 3) |
| | 5 | 0 = Operation is performed on Bus A<br>1 = Operation is performed on Bus B |
| | 6 | 1 = Interrupt processor upon successful bus operation |
| | 7-11 | Receive terminal address |
| | 12=16 | 00000, 11111 = mode command operation<br>00001<br>---<br>---<br>Receive terminal subaddress<br>11101<br>11110 = Asynchronous message |

| IW1 | BIT | DESIGNATION |
|---|---|---|
| | 1-5 | Word count or mode command code |
| | 6 | 0 = Command is not a broadcast command<br>1 = Command is a broadcast command |
| | 7-11 | Transmit terminal address |
| | 12-16 | 00000, 11111 = mode command oepration<br>00001<br>---<br>---<br>Transmit terminal subaddress<br>11101<br>11110 = Asynchronous message |

FIGURE 1.3.1-1
FORMAT FOR INSTRUCTION WORD PAIR




TYPICAL POINTER BLOCK USED BY BIU

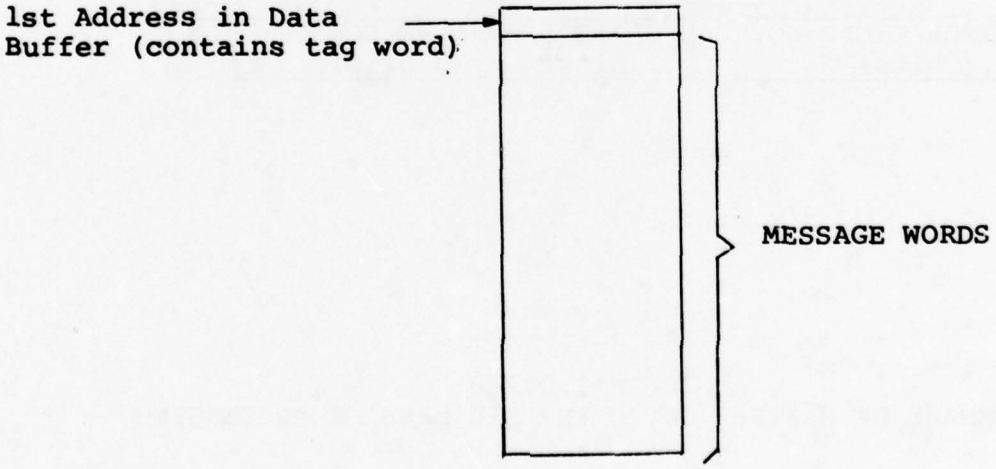



FIGURE 1.3.1-2
TYPICAL MESSAGE-DATA BUFFER

- BIU #2 DMA's instruction words 1, 2, from host
- BIU #2 generates command word
- BIU #2 appends $\overline{T/R}$ and subaddress bits of command word to the LS end of a 10-bit base address to form a 16-bit address into a pointer table:

| BASE ADDRESS (10 bits) | $\overline{T/R}$ | SUBADDRESS (5 bits) |
|---|---|---|

LSB

- BIU #2 DMA's pointer from pointer table; pointer is the location of the first address in the data buffer
- Pointer is stored in BIU #2's Pointer Register
- BIU #2 loads incremented value of pointer into external address register
- BIU #2 transfers command word to BIU #1
- BIU #1 handles data DMA's
- In the case of a RT-Transmit message, the final data DMA by BIU #1 is followed by a DMA by BIU #2 of the tag word into the first address of the data buffer. The tag word transferred contains the minor cycle number, word count and the data error bit:

| MINOR CYCLE NO. ( bits) | SPARE (2 bits) | WORD COUNT (5 bits) | DE |
|---|---|---|---|

LSB

FIGURE 1.3.1-3
SUMMARY OF MASTER CONTROLLER BIU MESSAGE PROCESSING

BIU #1 handles the DMA data transfers. If the message is an RT-Receive message, the data transfers by BIU #1 complete the message process. However, if the message is an RT - Transmit message, the data transfers by BIU #1 are followed by a final DMA transfer by BIU #2 of the tag word into the first address of the data buffer.

The discussion above assumed the BIU configured as a master controller. In the remote mode, command generation doesn't apply. However, buffer address generation does, and is as above except the command word used is that received from the manchester bus.

1.3.2 As mentioned above, the BIU sets interrupts on various conditions, including:

1) message errors,
2) status word exceptions,
3) certain mode commands, and
4) program requirements.

Interrupt generation reflects one of several facts:

1) The BIU has encountered data transfer problem and error indications cannot be overcome without host intervention, or
2) The BIU has been initialized because of a power dropout or startup and needs to be set up by the host, or
3) The BIU has finished the bus oriented tasks required of it by the BIU program in host memory and the program requires host notification, or,
4) The host decides to intervene in BIU operation and commands the BIU to gracefully halt operation.

As the word itself indicates, an interrupt is a break in an on-going operational screnario, and when such a break occurs, reasons for interrupt generation are recorded by the BIU in BIU #2's Internal Status Register (ISR) and Bit Register (BIT). Both of these are available to the BIU host, and BIT in a remote controller is available to the master controller via the manchester bus by use of a mode command.

These interrupts can be viewed from two perspectives, that of the master controller and the remote controller. These aspects are covered below.

- BIU #1 receives an RT-Transmit or an RT-Receive command.
- BIU #1 signals to BIU #2 that a command word is present and passes the command word to BIU #2 via the I/O lines.
- BIU #2 determines the command is an RT-Transmit or an RT-Receive command and begins data buffer address generation.
- BIU #2 appends the T/R and subaddress bits of the command word to the LS end of a 10-bit base address to form a 16-bit address into a pointer table.
- BIU #2 DMA's pointer from pointer table; pointer is the location of the first address in the data buffer.
- Pointer is stored in BIU #2's Pointer Register.
- BIU #2 loads incremented value of pointer into external address register.
- BIU #1 handles data DMA's.
- In the case of an RT-Receive message, the final data DMA by BIA #1 is followed by a DMA by BIU #2 of the tag word into the first address of the data buffer.

FIGURE 1.3.1-4
SUMMARY OF REMOTE CONTROLLER BIU MESSAGE PROCESSING

1.3.2.1 Considering the BIU as a master controller, interrupts are generated and actior is stopped when various conditions exist. These conditions may be of such nature that would be corrected by additional message communication attempts by use of the automatic re-try capability of the BIU. This tends to separate hard and soft failures. Interrupt conditions that occur which would not allow a re-try are as follows:

1) DMA transfer handshake failure
2) Power-on-Reset condition
3) Instruction Halt (from Fetch Sequence)
4) Specified status word bits
5) Invalid instruction set (from Fetch Sequence)
6) Abort (from Host)

Other conditions for interrupts which would allow a re-try attempt are as follows: Re-try Count = 0 and,

1) Message error (transferred from BIU #1)
2) Specified status word bits

Finally, the last conditions for interrupts are conditions sensed during a normal message providing none of the above occurred. These conditions are as follows:

1) Program Controlled Interrupt (from Fetch Sequence)
2) Mode Data Present (BIU receiving mode command data)
3) Halt Command (from Host)
4) Asynchronous Message (Command Address = 30)

1.3.2.2 When the BIU is operating in the remote mode, interrupts are generated for similar conditions as before. In this mode, interrupts are generated and action is stopped when any of the following conditions occur:

1) DMA Transfer handshake failure (BIU #1)
2) Power-on-Reset
3) Asynchronous Message (Command Address = 30)
4) Synchronized Mode Command
5) Mode Data Present (received Mode Command Data)
6) Master Function (BIU receives Mode Command 10001)

The BIU will remain inoperative until serviced by the host.

A. ERROR INTERRUPTS

NO RE-TRY ALLOWED

1) DMA transfer handshake failure
2) Power-on-Reset condition
3) Instruction Halt (from Fetch Sequence)
4) Specified status word bits
5) Invalid instruction set (from Fetch Sequence)
6) Abort (from Host)

B. ERROR INTERRUPTS

RE-TRY ATTEMPTS ALLOWED

Re-Try Count = 0 and,

1) Message error (transferred from BIU #1)
2) Specified status word bits

C. SYSTEM INTERRUPTS

1) Program Controlled Interrupt (from Fetch Sequence)
2) Mode Data Present (BIU receiving mode command data)
3) Halt Command (from Host)
4) Asynchronous Message (Command Address = 30)

FIGURE 1.3.2.1-1
BIU INTERRUPTS IN CONTROLLER MODE

A. ERROR INTERRUPTS

1) DMA Transfer handshake failure (BIU #1)
2) Power on Reset

B. SYSTEM INTERRUPTS

1) Asynchronous Message (command address = 30)
2) Synchronize Mode Command
3) Mode Data Present (received Mode Command Data)
4) Master Function (BIU receives Mode Command 10001)

FIGURE 1.3.2.2-1
BIU INTERRUPTS IN REMOTE MODE

1.3.3 When an interrupt is generated by the BIU, BIU #2 signals the host via its Interrupt Request line that the BIU needs service. Just before requesting service, BIU #2 sets the Busy/Continue bit in its Processor Control Register (PCR) and in BIU #1's Status Register. After generating the request, BIU #2 monitors the Busy bit waiting for a change of condition.

In a remote controller BIU any command sent to it is implemented by simply returning the status word with the busy bit set; no DMA activity occurs.

In either controller BIU, the activity allowed during the busy condition consists of:

1) monitoring the GO and Busy bits, and
2) implementing the control codes required by the host of BIU #2.

The control codes available to the host are:

| | | |
|---|---|---|
| 0000 | Load | Mode Data Register |
| 0001 | Load | Master Function Register |
| 0010 | Load | Instruction Address Register |
| 0011 | Load | Base Address Register |
| 0100 | Load | Processor Control Word |
| 0101 | Load | Status Word Data |
| 0110 | Load | Built-In-Test Register |
| 0111 | | Abort Command |
| 1000 | Output | Mode Data Register |
| 1001 | Output | RCV Status Word/MFR |
| 1010 | Output | Instruction Address Register |
| 1011 | Output | Base Address Register |
| 1100 | Output | Processor Control Register |
| 1101 | Output | Internal Status Register |
| 1110 | Output | Built-In-Test Register |
| 1111 | | Halt Command |

These codes give the host control of the BIU during control-code initiated or power-up or power-dropout initiated busy states.

When BIU #2 initializes itself, it sets the POR bit in BIT word. It then clears the ISR and other bits in BIT and sets the Busy and No-Go bits of the PCR to busy (0) and No-Go (0). Finally, it flags its condition by setting its interrupt line active. While in the initialized state, BIU #2's registers may be read or written. The PCR is somewhat special. When loading the PCR, both BIU # 1 and BIU #2 accept data from their respective I/O ports. The control code, Load PCR, sent BIU #2 together with the control code strobe causes BIU #2 to generate a code-execute strobe to BIU #1. Then, both BIU's load their respective PCR's. If the Processor Control Word from the host has Busy and No-Go bits set to busy and no-go, previcus state, BIU #1 will now respond to all valid commands with the status word showing the busy status bit set. To properly enter the BIU into an operational screnario, BIU #2's BAR, IAR and MFR should be loaded; then the PCR of both BIU #1 and #2 should be loaded. To remove the BIU from the active state to the busy state, control code 1111 (halt gracefully) should be sent to BIU #2. BIU #2 will complete message processing if a message is present, set the BIU into the busy state, and then interrupt the host. At this point any of the registers of interest can be read and reloaded if necessary. Finally, PCR's should be loaded to re-enter active operation.

1.4 In summary, the BIU chip set performs the functions required by MIL-STD-1553B, such as Mode Code execution, RT RT transfers, broadcast operation, and status word formats, in addition to the host interface action described herein. Figure 1.4-1 shows a typical chip-set connection diagram.

FIGURE 1.3.2-1
BIT WORD FORMAT

FIGURE 1.3.2-2
INTERNAL STATUS REGISTER (ISR)

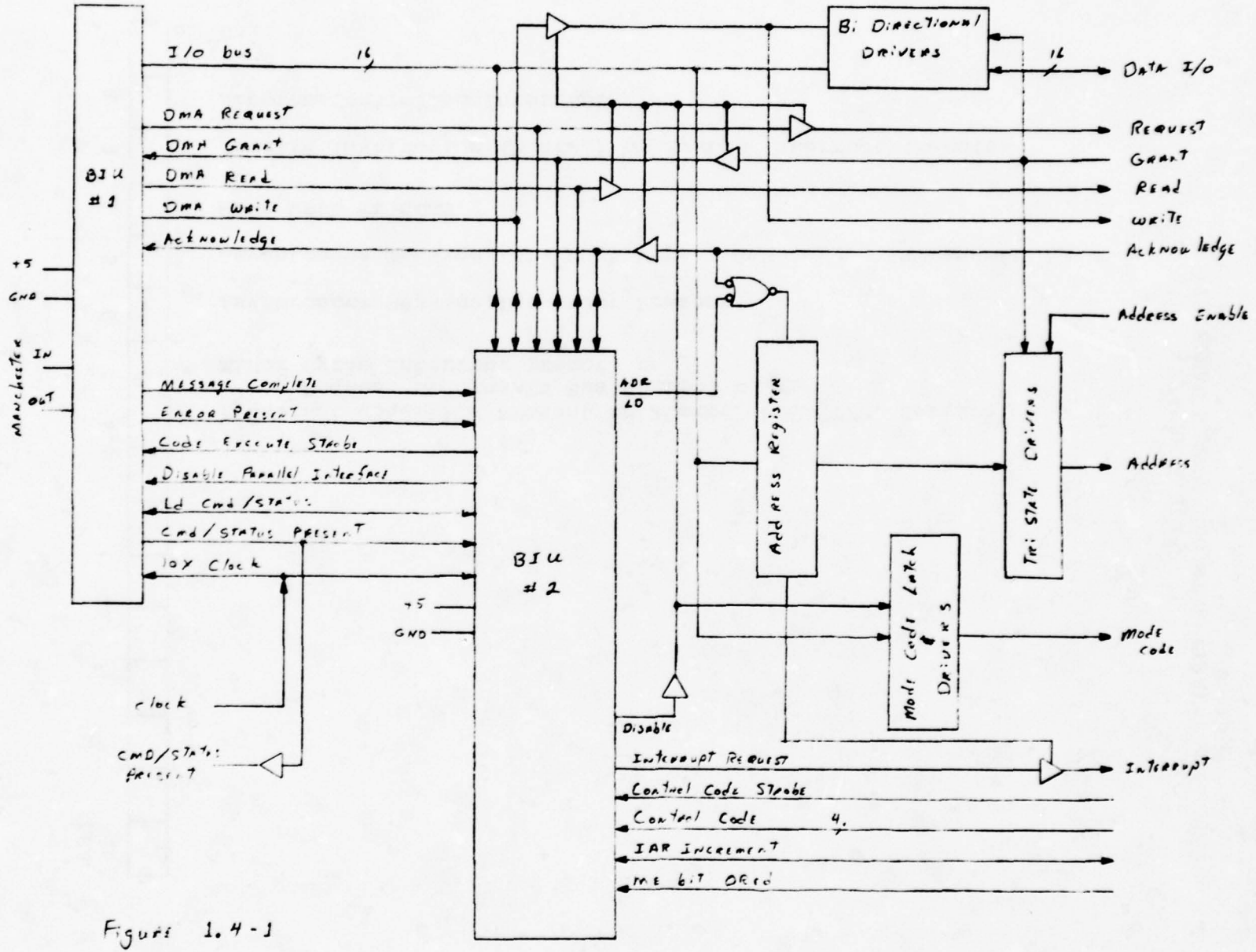

BIU #1
I/O bus
16
DMA Request
DMA Grant
DMA Read
DMA Write
Acknowledge
+5
GND
Manchester IN
OUT
Message Complete
Error Present
Code Execute Strobe
Disable Parallel Interface
Ld Cmd/Status
Cmd/Status Present
10X Clock
clock
CMD/STATUS PRESENT
BIU #2
+5
GND
ADR LD
Disable
Interrupt Request
Control Code Strobe
Control Code
4
IAR Increment
ME bit Offed
Address Register
Mode Code Latch & Drivers
Tri State Drivers
Bi Directional Drivers
16
Data I/O
Request
Grant
Read
Write
Acknowledge
Address Enable
Address
mode code
Interrupt

Figure 1.4-1

# SOME APPLICATIONS OF THE ROCKWELL DATA TERMINAL CHIP

B. E. Bona

Autonetics Strategic Systems
Anaheim, California

## 1. INTRODUCTION

The Data Terminal (DT) chip is a radiation hardened, micro-programmable device designed for use in Manchester encoded serial data buses. This paper briefly describes the chip functions and gives some typical examples of its use. The examples include a MIL-STD 1553A interface to a doppler velocity system, application in a round-robin bus structure where the broadcast mode is used, and in complex dual redundant bus whose protocol is similar to MIL-STD 1553A, but with longer word length (i.e., B-1).

## 2. DATA TERMINAL FUNCTIONS

The DT chip is a large scale integrated CMOS/SOS circuit[1] designed to perform the logic and digital functions required to mechanize a Manchester encoded time division multiplexing system, as described, for example, in MIL-STD 1553A. The DT circuit can be used in either the remote terminal or controller modes. The DT circuit was designed to meet the requirements of MIL-STD 1553A and, therefore, utilizes the fixed synchronizing patterns defined therein. The DT circuit, however, is programmable so that variations on word length (e.g. B-1 and Space Shuttle), field definition, and bus protocol can be accommodated with appropriately programmed external ROM(s). A brief summary of the functions provided by the DT circuit is as follows:

(1) Manchester encode and decode.

(2) Sync detection (Receive) and Sync generation (Transmit).

(3) Terminal address compare.

(4) Output and input NRZ in sync with data clock.

(5) Store and output word count

(6) Check parity (Receive) and generate parity (Transmit).

(7) Error detection: valid Manchester; word length; parity; message length.

(8) Assemble and transmit status and data words. (Includes sync signals and parity).

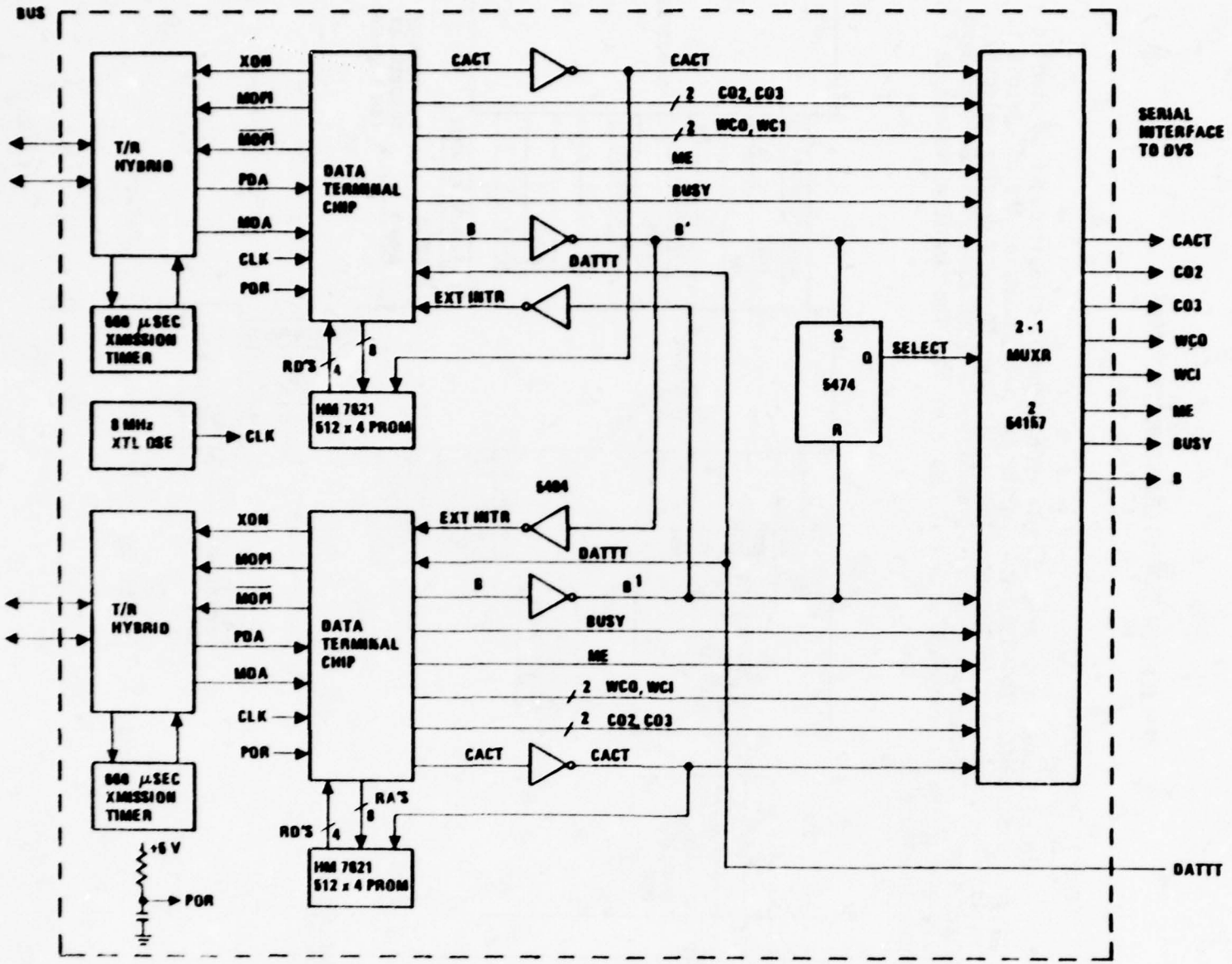



Figure 3. Dual Redundant Data Bus Circuit

## 3. APPLICATION IN DOPPLER VELOCITY SYSTEM

### 3.1 General Description

Figure 1 shows a generic example of how the DT circuit might be used in a single channel MIL-STD-1553A bus. In the following example, the DT circuit is used in a dual redundant 1553A bus. The application is for a doppler velocity system (DVS). The communication is rather simple: each time a transmit command is received, the DVS responds with three data words. The complete message is shown in Figure 2.

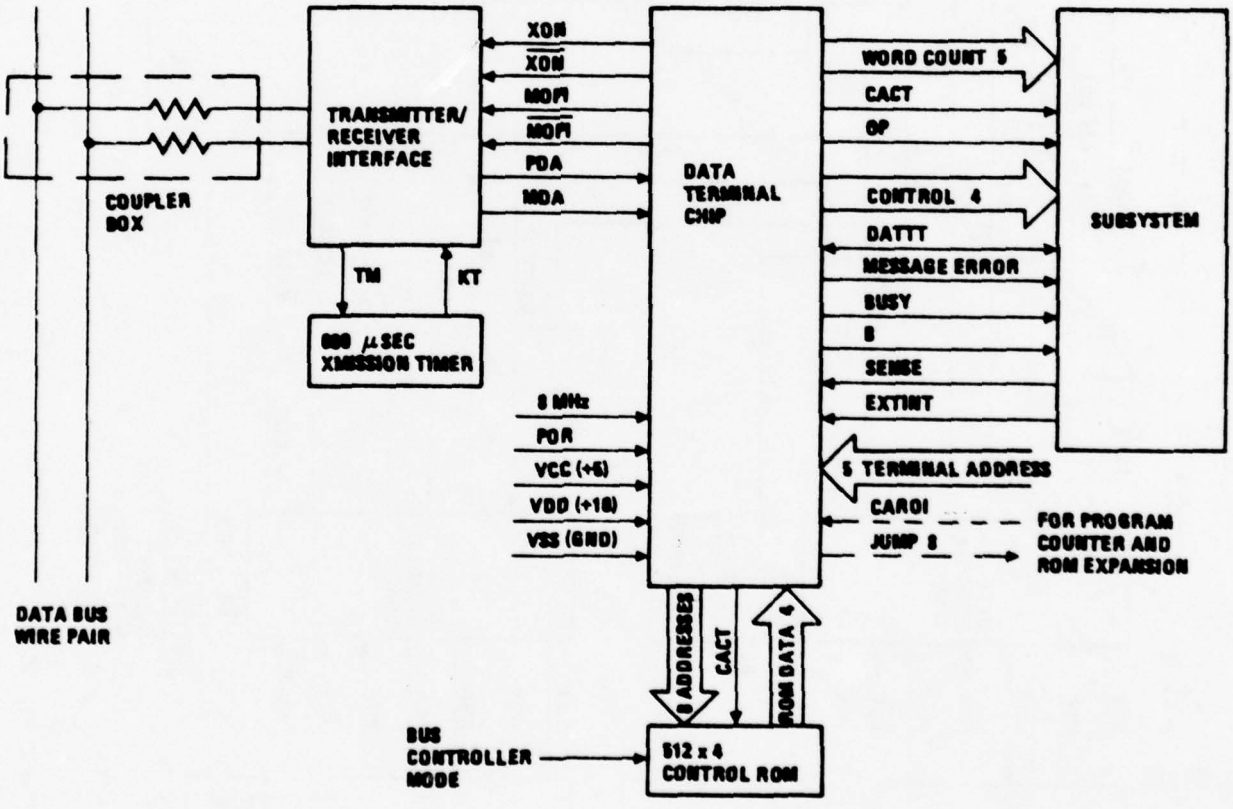



Figure 1. Remote Terminal/Controller Configuration

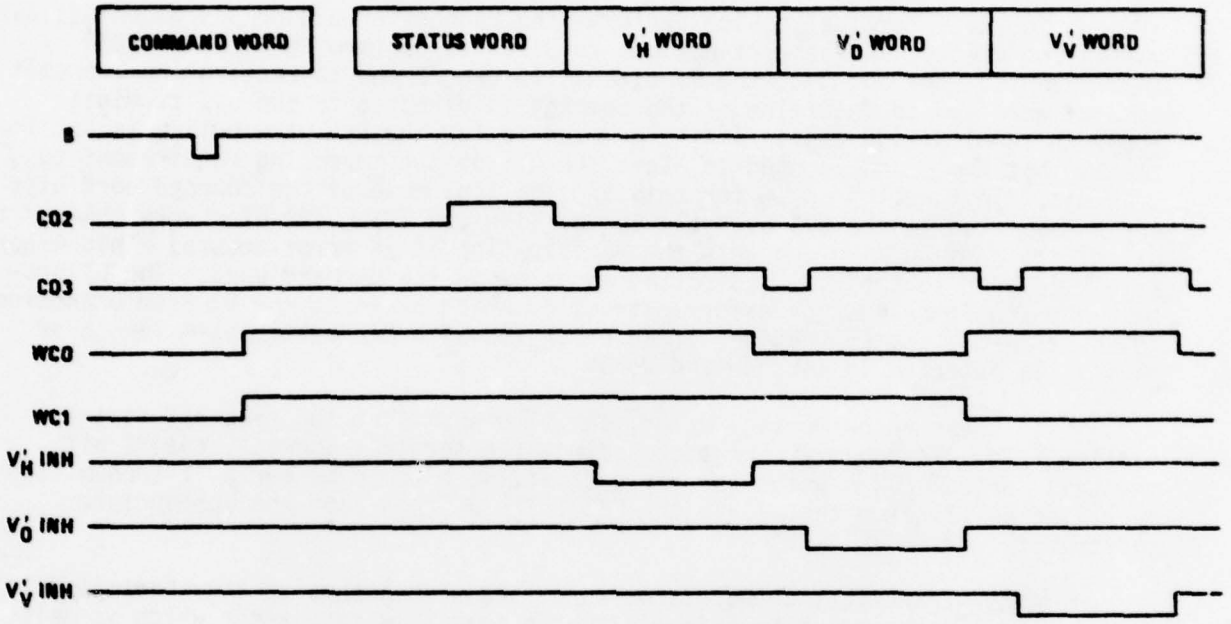



Figure 2. Timing for Data Transfer

The dual redundant MIL-STD 1553A data bus circuit (DBC) used to interface with the DVS is shown in Figure 3. The circuit consists of two independent bus channels. Each of these channels is configured with a T/R hybrid unit and with a data terminal device along with its associated ROM(s), as shown in Figure 3.

Outputs of two primary channels are cross-coupled in such a way that the channel receiving the most recent command takes precedence. The interface from the DBC to the DVS is bit-serial over the bi-directional data line DATTT, at a clock rate of 1 MHz. Clock (CACT), control (CO's), word count (WC), message error (ME), early warning (B), and busy lines are also used to interface the DBC with the DVS.

## 3.2 Word and Message Control

The data terminal device performs the Manchester coding and decoding, data validation checks, and other functions required to conform to MIL-STD 1553A requirements. For the DVS, the DT circuit is programmed to receive the Transmit command word and to determine if the command is directed to the DVS terminal (compare terminal address). If the command is for the DVS, the B line is set low for two bit times and is used to alert the DVS of the impending requirement to transmit. In the DT program for this application, none of the command word bits are transmitted to the DVS over the serial line (DATTT). The DT checks the parity bit of the command word and sets the ME flip-flop if an error occurs. This error signal is sent to the DVS, indicating an error in the command word. The DT program has a halt-on-message error instruction which prevents the DT from branching to the transmit status routine, if an error (invalid Manchester word length or parity) is detected in the command word.

If there is no message error, the DT branches to the transmit status routine. The DT transmits the status sync, the terminal address, the ME bit, and sets control CO2 causing the ten data status bits to be serially transmitted over line DATTT, from the DVS to the DBC. The DT then adds the appropriate parity bit.

After transmitting the status word, the DT branches to the transmit data routine where it proceeds to transmit the three data words, after which it halts. Word count lines WCO and WC1 and control line CO3 are used to strobe the velocity data from holding registers located in the DVS. Figure 2 shows the basic timing associated with the data transfer. Figure 4 shows some typical interface logic that might be implemented in the DVS. The signals $V_v$inh, $V_d$inh and $V_h$inh are used to load data into the registers. Loading is accomplished when these signals are high; data is serially transferred when they are low.

## 3.3 Channel Control

The channel that interfaces with the DVS system is controlled by the combination of a channel select flip-flop and a multiplexer. When a channel detects a command sync, it sets the DT program counter to the location hex 80. If the terminal addresses compare, the B flip-flop is set to the T/R bit (T/R = 1 for DVS), and remains set for two bit times. This signal sets/resets the channel select flip-flop which in turn couples the appropriate DT outputs to the DVS via the two 54157 multiplexers. The signal is also used to halt the other DT via the external interrupt line.

## 3.4 DVS Program Description

Flow charts and program listings for the transmit command, status, and transmit data routines are shown in Tables 1 through 3 and Figures 5 through 7.

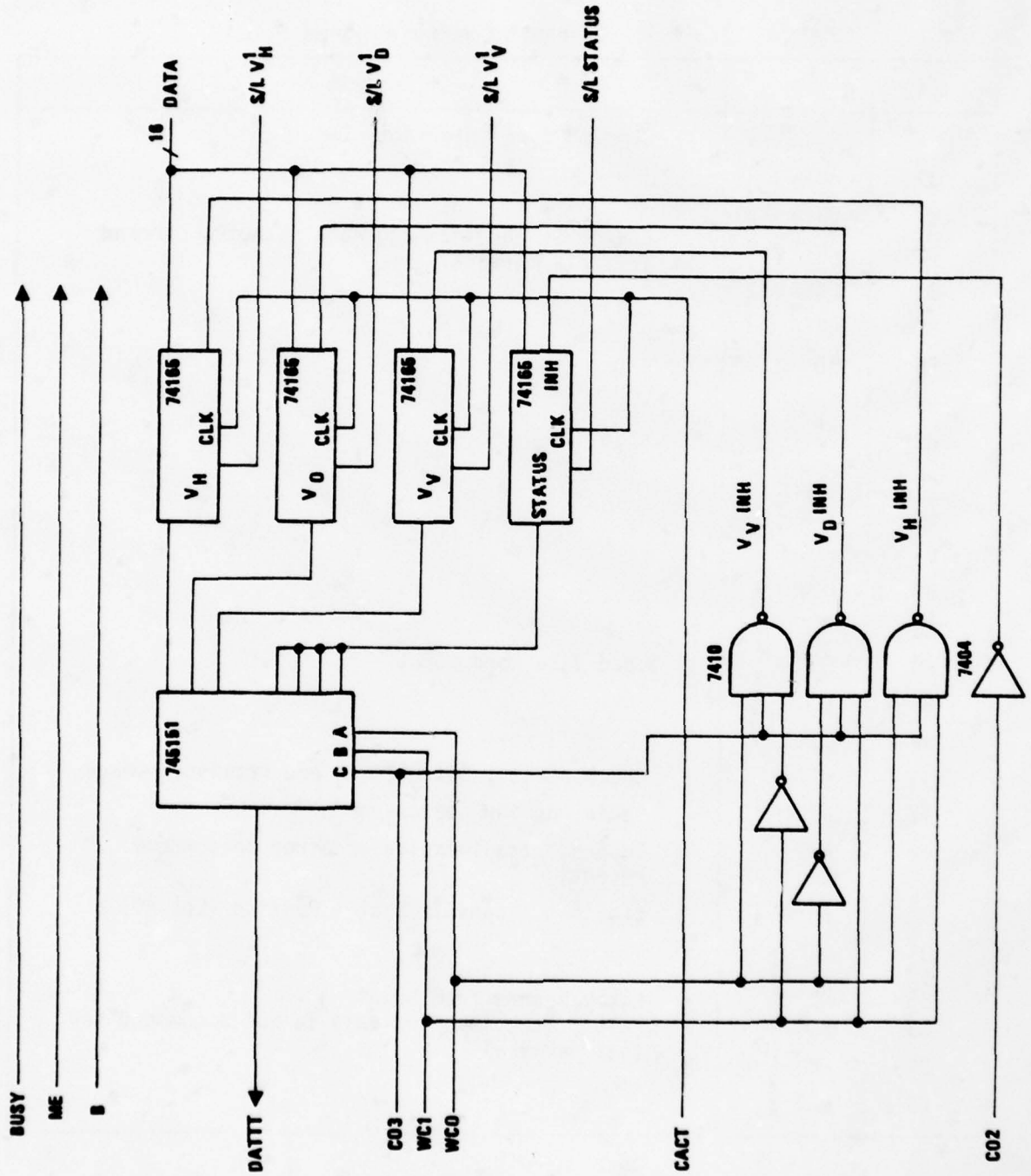



Figure 4. Typical DVS Interface Logic

Table 1. Transmit Command Program

| IA | I | Transmit |
|---|---|---|
| - | - | Command Sync Interrupt |
| 80 | BNTA 0 | Branch if the incoming data is not a command for this terminal |
| 81 | BNTA 0 | |
| 82 | BNTA 0 | |
| 83 | BNTA 0 | |
| 84 | BNTA 0 | |
| 85 | HB | |
| 86 | FLG F | |
| 87 | RB | Reset B |
| 88 | FLG F | |
| 89 | FLG F | |
| 8A | FLG F | |
| 8B | LDWC | Load data word count |
| 8C | FLG F | |
| 8D | FLG F | |
| 8E | FLG F | |
| 8F | LDWC | |
| 90 | PTY 0 | Check parity; Set CO's to end receive command. |
| 91 | FLG F | Begin gap for status word |
| 92 | HME | Suppress transmission if error on command reception |
| 93 | BR A | Branch to Transmit Status Routine (Loc A0) |
| 01 | HLT | Entry Points from 80-84<br>Halt if the incoming data is not a command for this terminal |
| 02 | HLT | |
| 03 | HLT | |
| 04 | HLT | |
| 05 | HLT | |

Table 2. Transmit Status Routine

| IA | I | Transmit Status |
|---|---|---|
| A0 | THI 0 | Transmit a command/status sync; set CO's for begin transmit status |
| A1 | TONE | |
| A2 | TLO 0 | |
| A3 | TTA | |
| A4 | FLG F | Transmit the address of this terminal |
| A5 | FLG F | |
| A6 | FLG F | |
| A7 | TTA | |
| A8 | TME 0 | Transmit ME; do not Change CO's |
| A8 | FLG F | Input and transmit status codes |
| : | : | |
| B1 | FLG F | |
| B2 | FLG F | |
| B3 | PTY 0 | Transmit parity, reset CO's |
| B4 | [1st Instruction of Transmit Data] | Begin Transmit data routine at this point; no branch permitted. |

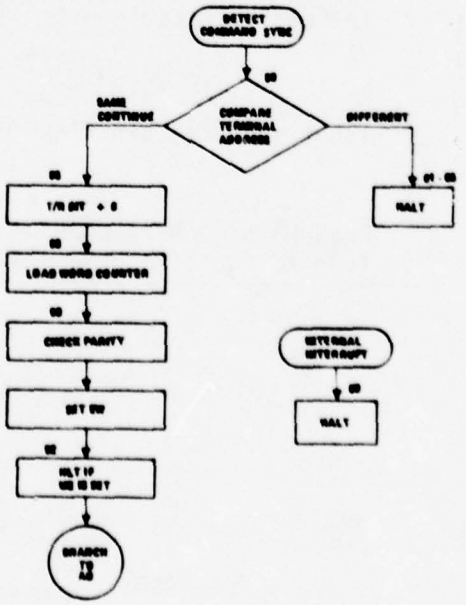



Figure 5. Transmit Command Flow Chart

Table 3. Transmit Data Routine

| IA | I | Transmit Data |
|---|---|---|
| $B4_{16}$ | TLO 0 | Transmit data sync and set CO's for begin transmit data |
| B5 | TZRO | |
| B6 | THI 8 | |
| B7 | FLG F | Input and transmit first 14 bits of data word |
| : | : | |
| C0 | FLG F | |
| : | : | |
| C4 | FLG F | |
| C5 | DDWC | Decrement the data word count |
| C6 | BCNZ E | Branch if this is not the last data word |
| C7 | PYT 0 | Transmit parity; reset CO's |
| C8 | HLT | Halt; operation complete |
| E0 | PTY 0 | Entry point for branch in C6; transmit parity; reset CO's. |
| E1 | TLO 0 | Transmit data sync for next data word; set CO's for begin transmit data |
| E2 | TZRO | |
| E3 | THI 8 | |
| E4 | FLG F | Input and transmit first part of next data word |
| : | : | |
| EB | FLG F | |
| EC | BR C | Branch into corresponding point of original transmit data sequence (above) |

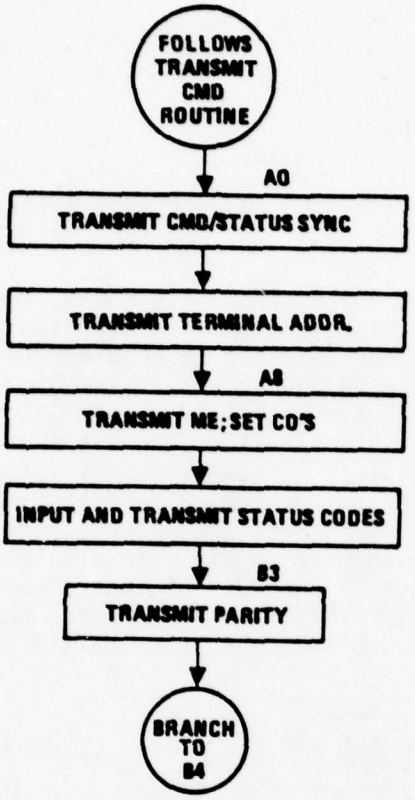



Figure 6. Transmit Status Flow Chart

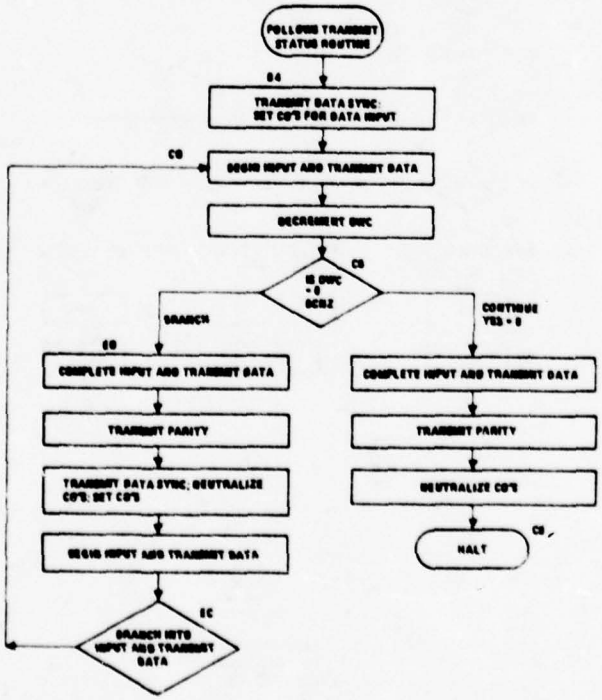


Figure 7. Transmit Data Flow Chart

## 4. ROUND-ROBIN/BROADCAST FUNCTIONS

In systems that employ broadcast modes, a remote terminal is required to respond to its own unique address, in addition to a broadcast address. In Round-Robin systems, a terminal may be required to respond to some of all of the other terminals sending messages(2). The DT device can be easily programmed to satisfy the above requirements. If a terminal is required to respond to n addresses, then basically n parallel channels are needed. Figure 8 shows a program with three channels. If the terminal address for Channel 1 is xxxxx, then Channel 2 will respond to terminal address xxxxxx and Channel 3 to address xxxxx. It is interesting to note that each channel can be programmed to process the data following the terminal address in a unique way. This feature should be extremely useful in Round-Robin systems, where each channel would correspond to a unique transmitting terminal. Note, however, that this additional versatility is accompanied by the requirement to provide a distinct and separate ROM program for each terminal.

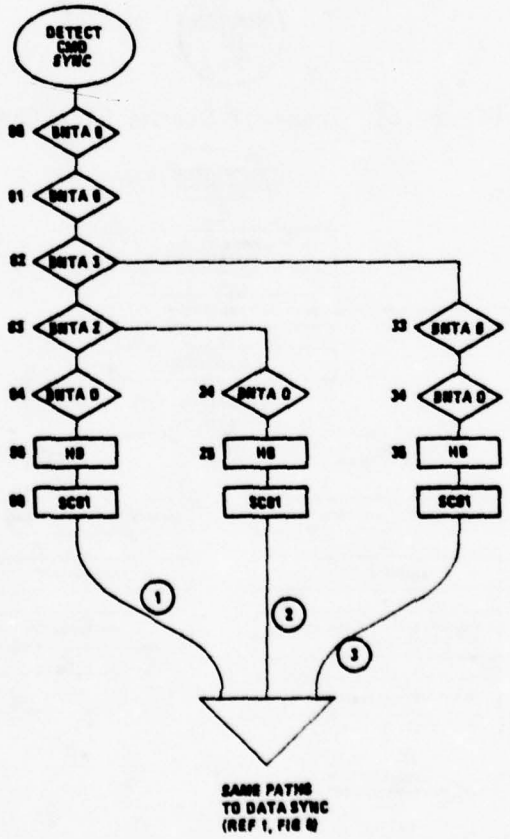



Figure 8. DT Flow for Multiple Terminal Addresses (XXXXX, XXXX̄X, XX̄XX)

In MIL-STD 1553A systems with one broadcast address (e.g., 11111), it is possible to have one ROM program for up to sixteen terminals. Figure 9 shows a program that utilizes the Branch on Data (BRD) instruction to implement the broadcast mode for address (11111), where the sixteen terminal addresses are of the form (oxxxx).

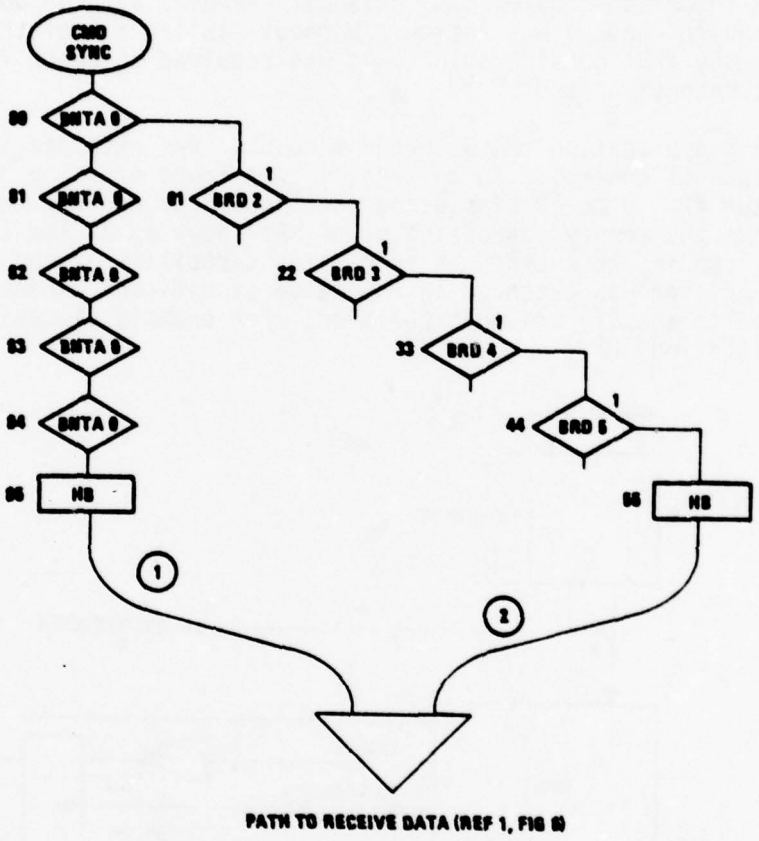



Figure 9. Unique Address (oxxxx) Channel 1 and Broadcast Address (11111) Channel 2

The program in Figures 8 and 9 are only used to illustrate versatility of the DT programming features. As with any programmable device, there are generally numerous programs that will accomplish the same objective. Such is the case with the DT device.

## 5. B-1 MULTIPLEX-INTERFACE-MODULE

The DT device was intended for use in the B-1 EMUX system before its demise. The B-1 protocol was similar[3] to MIL-STD 1553A, but with significant differences, a few of which include a 24-bit word length, no terminal-to-terminal mode, and four leading validity bits followed by 16 bits as specified in 1553A. The multiplex-interface-module (MIM) had dual channels with unique requirements for defining which channel was active. Without listing all of the requirements, suffice it to say that considerable logic was required to check for valid commands and to effect channel control.

In this application the DT program counter was extended to accomplish much of the channel control. To do this, it was found expedient to extend the DT program counter from 9 to 10 bits using the lines designed for this purpose. Figure 10 shows the memory addressing used. As shown an 8k ROM (10 × 8) was used, with a flip-flop used to extend the addressing capability of the DT to 10 bits. The address register was extended in the lease significant direction and a binary point was used to specify this bit position. For example, location hex 80 is extended to location 80.0.

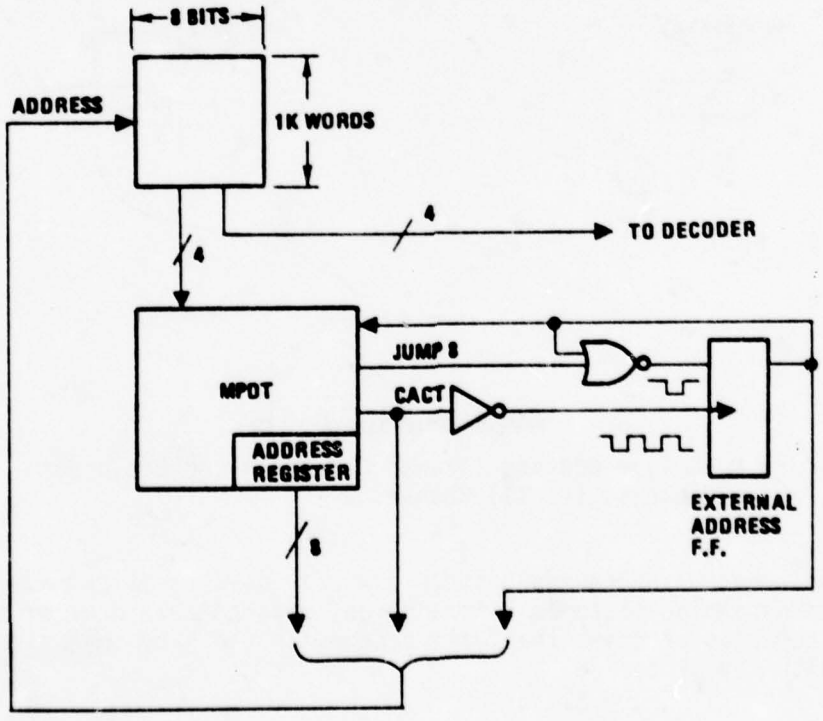



Figure 10. Memory Addressing for B-1

The DT requires four data lines from the ROM. The four extra data lines from the 8k ROM were used to extend the basic DT instruction set. As indicated in Figure 10, four ROM data lines are directed to the DT and the other four are decoded to generate the additional control instructions. Normally the DT uses one instruction per bit time, but with the extension shown in Figure 10, it is possible to perform two instructions per bit time. In the B-1 application, these extended instructions were used to accomplish channel control, load external registers, and perform assorted other logic functions in synchronism with the NRZ data and system clock (CACT).

## 6. SUMMARY

This paper has presented some application of the DT device. As seen from the discussion, the device can be used from the simplest to the most complex system with relative ease and efficiency. As with any programmable device, its use is limited only by the imagination of the user.

## REFERENCES


1. B. E. Bona, "Micro-Programmable Data Terminal," Multiplex Symposium, Dayton, Ohio, November, 1976

2. G. Beasley, "Round-Robin System" Multiplex Symposium, Dayton, Ohio, October, 1978

3. B. D. McLanen, "B-1 Avionics Data Multiplexing System (AMUX), Multiplex Symposium, Dayton, Ohio, November, 1976

# A HYBRIDIZED 1553A INTERFACE

W. R. Hunter
Teledyne Systems Company
Northridge, California

## ABSTRACT

This report describes a 1553A interface designed to enable a Teledyne computer to function as Bus Controller and/or Remote Terminal in a dual-redundant Multiplex Data Bus system. The interface complies with MIL-STD 1553A requirements and is designed to connect directly to most general purpose computers. For minimum size and weight, all circuit components are in dice form fabricated in hybrid packages. There are three functionally unique hybrids:

- A Driver/Receiver hybrid
- Multiplex Terminal Unit (MTU) hybrid
- A Device Control Unit (DCU) hybrid

The Driver/Receiver hybrid couples directly to a dual data bus system accommodating TTL control and data signals on one side and +10V Manchester bi-phase signals on the bus side. The MTU provides a full-duplex serial interface between the Driver/Receiver and DCU hybrids. MTU functions include code conversion between NRZ and Manchester, serial timing and formatting for input/output data, validity checks on received data, and a total message length monitor. The DCU performs all message handling requirements of a Bus Controller and/or Remote Terminal. It utilizes the computer's main memory for working storage moving data in and out via Direct Memory Access (DMA).

Computer software presides over DCU Bus Controller operations. Each bus transaction is initiated by outputting a message pointer and a control word specifying the number of transmit/receive words. As a Remote Terminal, the program specifies RT address and controls 10 bits of the status word. Controller and Remote Terminal functions are assigned separate DMA ports thereby enabling simultaneous BC/RT operation for 100% self-test capability. Other design features include:

1. Bus Controller memory addressing up to 65K; separate Remote Terminal memory addressing of 2K.
2. A "next message queue" for continuous Controller operations.
3. Remote terminal-to-remote terminal messages.
4. A message error status register with interrupt and an automatic system reset on excessive message length times.

5. A dead-line discrete indicating no bus traffic.

6. A buffer to hold the last received command as remote terminal for program interrogation.

## INTRODUCTION

MIL-STD 1553 rigidly defines the communications protocol and data structure for transferring information over a time multiplexed bus. This results in a high degree of standardization for interfacing different equipment types as well as a cost effective means for interconnecting numerous dispersed subsystems. Design complexity of each interface depends on the subsystem being interfaced and the performance features required by the end-user. In any design, however, those areas most challenging are:

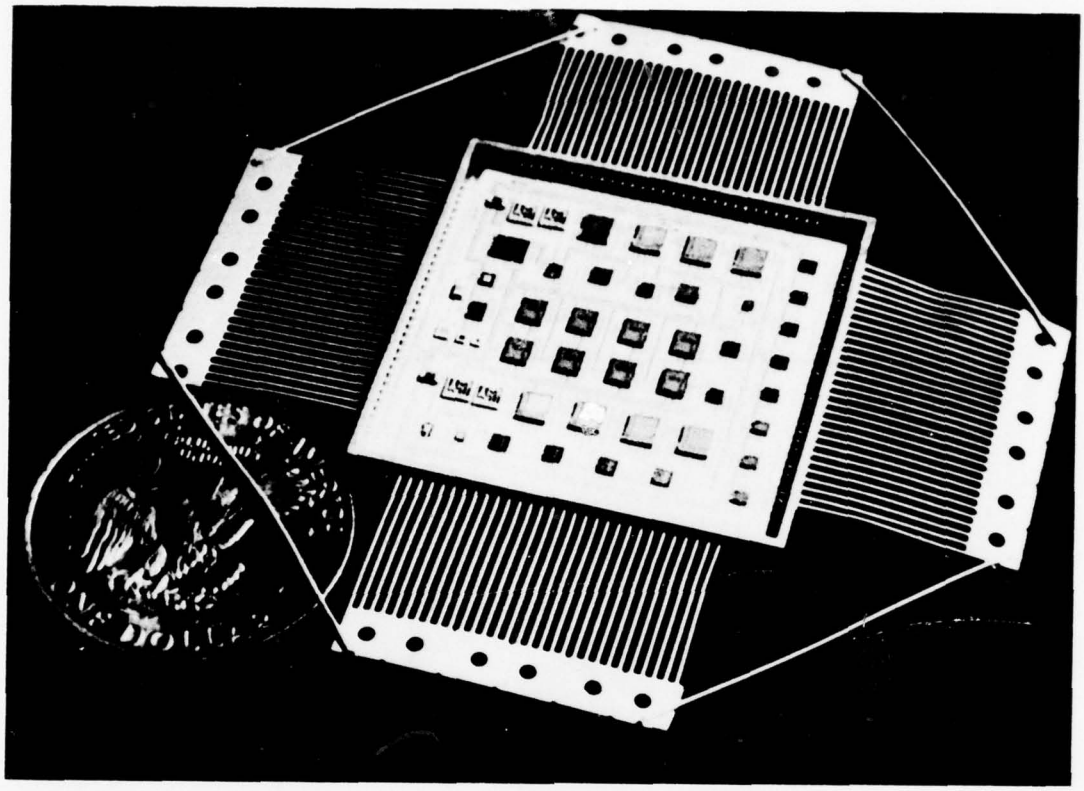
- Reliable extraction of data off the bus
- Predictable recovery from all error situations
- Software control versus dedicated hardware solutions
- Built-in self test
- Designing for the eventualities of MIL-STD 1553B

Teledyne's 1553 Interface is designed to mate with most general purpose computers having direct memory access (DMA) and programmed I/O capability. DMA operations are 16-bit parallel structured around popularly used core memory timing. The programmed I/O functions utilize a 16-bit bi-directional data bus. The interface runs primarily off computer supplied clock for synchronous data exchange while front-end logic requires an accurate 8 MHz clock for timing data on and off the bus.

For size and weight considerations, all ciruit components are in dice form and custom fabricated in hybrid packages. A typical hybrid is shown in Figure 1. As shown in Figure 2, the 1553 hybrid trio consists of Device Control Unit (DCU), Multiplex Terminal Unit (MTU), and Driver/Receiver. With these elements, a host computer can operate over a dual data bus configuration as Bus Controller or as a Remote Terminal by exercising the interface controls summarized in Table 1.

## DEVICE CONTROL UNIT (DCU)

The DCU performs those functions required of a bus controller and remote rerminal as specified in MIL-STD-1553A for message processing. It is responsible for all message contents into and out of the local subsystem's own memory. The connected subsystem (computer or controller) directs the activities of the bus controller portion of the DCU by sending it starting addresses of the message in its memory and the number of words (commands and data) to be transmitted plus the number of words to be received (status and data). The remote terminal portion of the DCU operates without subsystem intervention other than specifying the remote terminal's address and 10 least significant bits of STATUS.

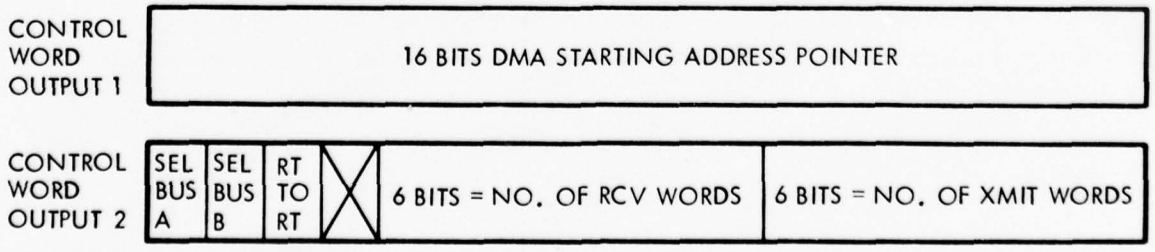


P107105

Figure 1. Typical Hybrid

The five most significant bits of the 16 bit RT DMA address select a 2048 word block assigned to buffer messages when the DCU is operating as a remote terminal. These bits are available on external pins. The DCU described below and shown in Figure 3 will operate independently and simultaneously as bus controller and/or remote terminal. That is, the bus controller can talk to itself as though it were 1 of 32 remote terminals providing a 100% closed-loop self-test capability.

To initiate a message-on the bus, the local subsystem outputs two control words to the DCU. One specifies a starting address in the subsystem's memory and the other defines the number of words involved. These control word formats are:




T119683A

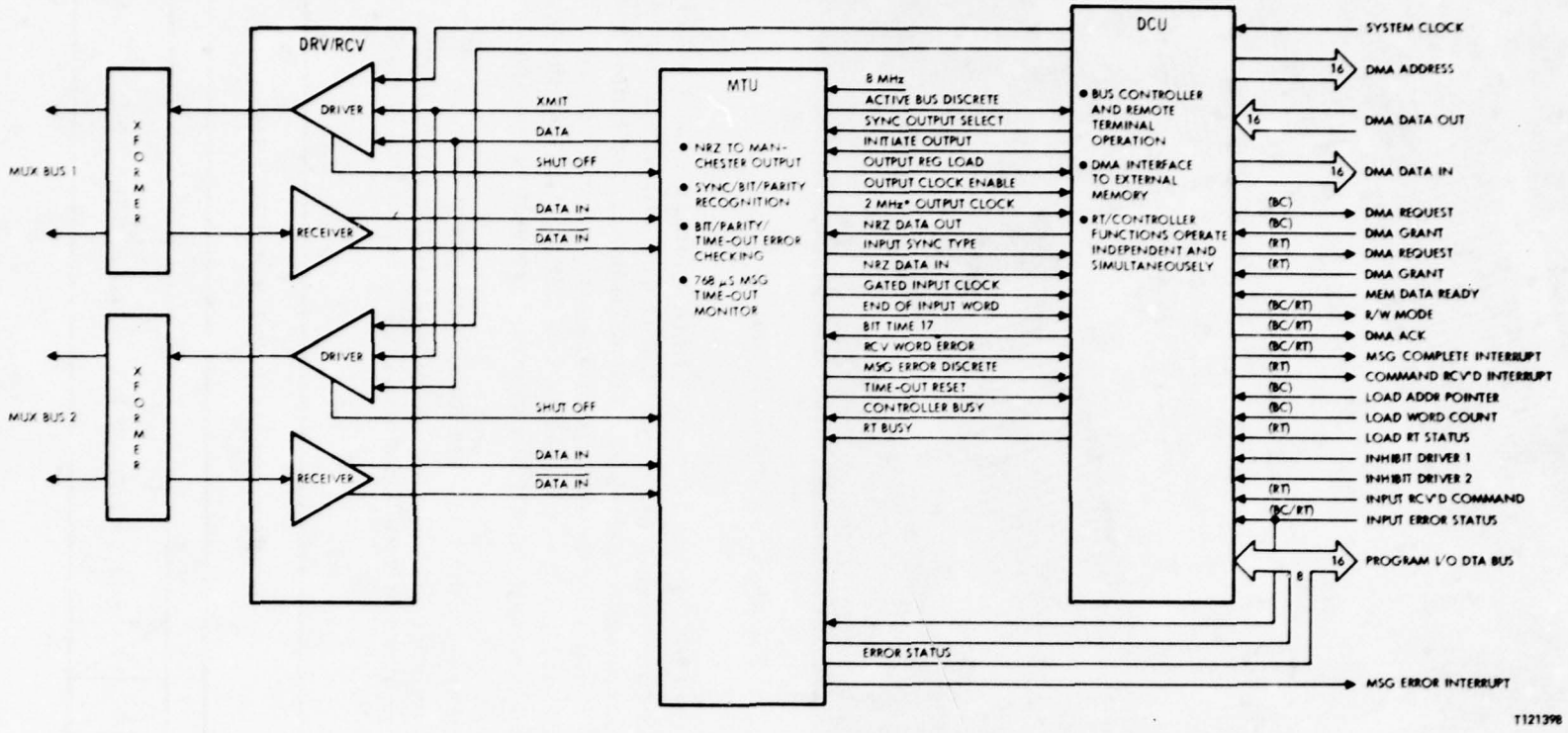



Figure 2. 1553A Hybrid Interconnect

Table 1. 1553 Interface Control

| Programmed Outputs | Programmed Inputs | Interrupts | Discretes |
|---|---|---|---|
| BC Address Pointer | MSG Error Status | BC/RT MSG Complete | BC Busy |
| BC Word Count/Bus Select | RT Commands | BC/RT MSG Error | RT Busy |
| RT Status/Address | | RT Command Rcv'd | Bus Inactive |
| | | | Inhibit Bus A |
| | | | Inhibit Bus B |

BC = Bus Controller

RT = Remote Terminal

1121735

If not currently busy, the DCU will immediately respond to output control word 2 by fetching (via direct memory access) the first command word of the message at the starting location specified by control word 1 and engage the MTU by raising the output sequence enable line. The sync type specified will be a command sync since the first word of each controller message is a command to some remote terminal. There are two word counters in the DCU, one for transmit and one for receive that are loaded with the contents of control word 2. The transmit word counter decrements each memory cycle until it reaches zero whereupon the DCU mode changes from transmit to receive and subsequent word inputs from the MTU are written contiguously into memory starting from the last output word location. The receive mode continues until the receive word counter reaches zero. At this time a message complete interrupt is issued. The DCU then determines if a new pair of control words has been placed in queue during the last message. If so, a new message commences. If not, the DCU goes idle.

Memory allocation for bus messages is very flexible. The subsystem program may designate any length blocks (0 to 63) starting anywhere in 65K of memory. Data within each block is accessed sequentially. For example, bus controller to remote terminal message block would contain the remote terminal receive command, followed by the data, followed by a location reserved for receiving terminal status. The transmit word count would include the command word plus the number of data words. The receive word count would be one for the status reply. The word count in the output control word thus includes all message words — commands, data, and status. The 6-bit field accommodates the longest possible message type which is remote terminal to remote terminal. For this case two command words are transmitted, followed by a status word and up to 32 data words, and then a second status word.

The total message length time-out error logic in the MTU can be checked by specifying excessive words in the transmit or receive message portion. Two bits of control word 2 specify on which of the two busses the controller performs transactions: Bus A or Bus B, both A and B, or none. Bus switching times coincide with the start of each message under DCU control.

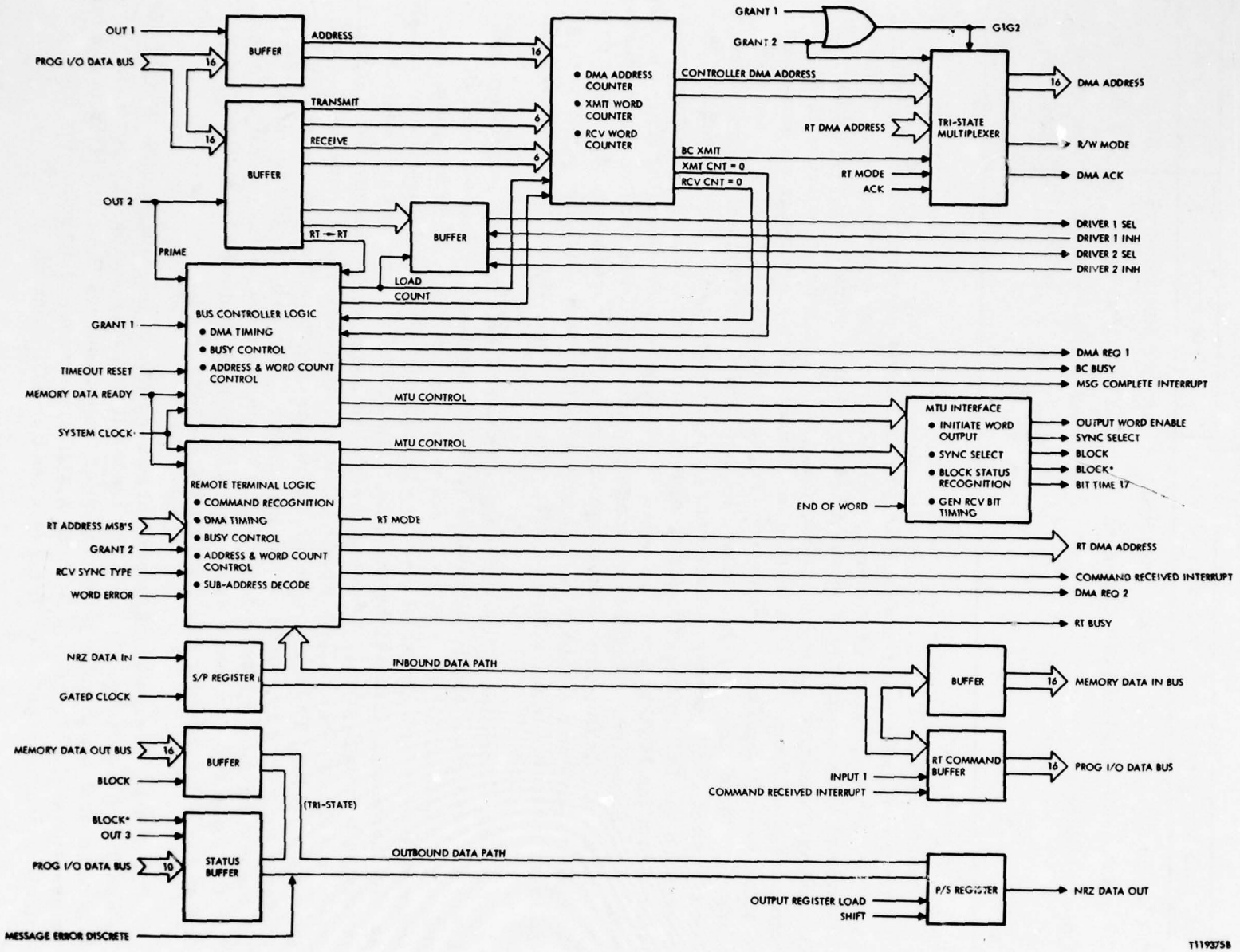



Figure 3. 1553 DCU Block Diagram

One bit in control word 2 is dedicated to the remote terminal to remote terminal mode of operation. The DCU uses this bit to cause the first two words to be output with command syncs.

The remote terminal (RT) portion of the DCU operates independent of the bus controller portion. It interfaces to the subsystem memory via its own DMA port working into and out of a 2048 word block specified by the subsystem. It requires no program control from the subsystem to function except for having its programmable ID register loaded. The program also has access to the 10 LSBs of the RT status word.

Serial input from the MTU is constantly monitored for a command word having a remote terminal address (ID) matching its own. The serially received bits are converted into a 16-bit parallel word and the END OF WORD pulse from the MTU is used as the time to examine the terminal address field (if the word was also preceded by the command sync signal). If a match is found, the logic will strobe the word count field of the command word into a counter, latch the subaddress field of the command along with the transmit/receive (T/R) bit, load the entire 16-bit command into a holding register, issue a command received interrupt to the subsystem and post the RT busy signal.

If other than a special function command (subaddress field $\neq 0$), the DCU takes one of two courses:

1. If the T/R bit = 0, the DCU will accept all the words that follow having a data sync preface and store them sequentially into the memory block specified by the subaddress field. When the number of memory cycles performed matches the received word count, the DCU places its status word out on the bus and when transmitted, exercises the same message complete interrupt used by the bus controller logic.

2. If the T/R bit = 1, the DCU immediately transmits its status word followed by the number of data words specified in the word count field, then generates the message complete interrupt.

Receipt of a special function command (subaddress field = 0) with the proper terminal address and T/R = 1 will cause the DCU to latch the command, generate the command received interrupt, and respond with status only. No further action is taken by the DCU and no busy is posted.

The RT logic can be redirected mid-message (if receiving) by another command with no consequences. The DCU will ignore all received commands that have detected bit or parity errors. The bus command that elicits remote terminal response is held until a new command replaces it. The program has ample time to interrogate the command register by executing the proper input instruction.

Only data words are transferred to/from memory in the RT mode. Commands and status words are not. In an RT-to-RT transfer, the DCU will pick out its

command ignoring all others and properly handle the data and status word requirements as specified in MIL-STD-1553A. The status word is supplied by hardware, i.e., not DMA'd from memory. The program, however, controls 10 bits of the status for whatever future systems may need. The MTU supplies the message error bit and the terminal ID is programmable.

Memory locations for remote terminal messages are normally specified by the subaddress field in the received command. Up to sixty-two 32-word blocks (2K) can be addressed using the T/R bit as part of the address. A subaddress field of all zeros is not used as a working RT address block in either transmit or receive. A transmit and receive command with the same subaddress fields will have RT memory blocks that are 1K apart. The data words of the message will be fetched or stored sequentially starting at the first location of the block. The five most significant bits of remote terminal DMA address are adjustable via hardwire so that the location of the 2K RT transmit/receive memory block can be positioned as system needs dictate.

## MULTIPLEX TERMINAL UNIT (MTU)

The MTU interfaces the DCU to the bus drivers and receivers by performing all bit and word serialization on data to and from the bus. It accepts Manchester encoded data from the line receivers and outputs NRZ data to the DCU. Similarly it takes (on command) serial NRZ encoded data from DCU and supplies Manchester encoded data to both line drivers. Input and output occurs independently and simultaneously, thus providing a full duplex serial interface to the DCU. The MTU also performs bit error checking, word parity checking, and a total message length time check as indicated in the block diagram of Figure 4. A dead-line discrete is also provided indicating no traffic on the bus for a time exceeding 65.5 milliseconds.

The MTU constantly monitors for bus activity. It is up to the system's bus controller to prevent both busses being active at one time. An "active bus" discrete signal is sent to the DCU by the MTU to specify which bus is active and providing data. This signal is intended for operating in the remote terminal mode so that any reply will be transmitted back on the active bus. If the DCU is active as a bus controller, then the active bus is specified by the local subsystem via a control word.

The TRUE/$\overline{\text{TRUE}}$ signals from each receiver are sampled with an 8 MHz clock in to a 24-bit register presenting a 3 $\mu$s snap-shot of incoming signal patterns. (A front end "transition stretcher" helps offset differing duty cycle relationships between receiver signals.) Eighteen bits of the 24-bit serial register are selected as a sample bit-pattern to address PROM memory coded for Manchester data bit recognition and 1553A sync pattern recognition. Recognition of a commnd or data sync (at any time) initializes all receiving logic and the search for valid data bits commences. A window is generated $\pm$250 ns about the theoritical mid-point of each data bit during which the bit recognition outputs of the PROM are examined. If a valid bit pattern is detected, the window re-syncs

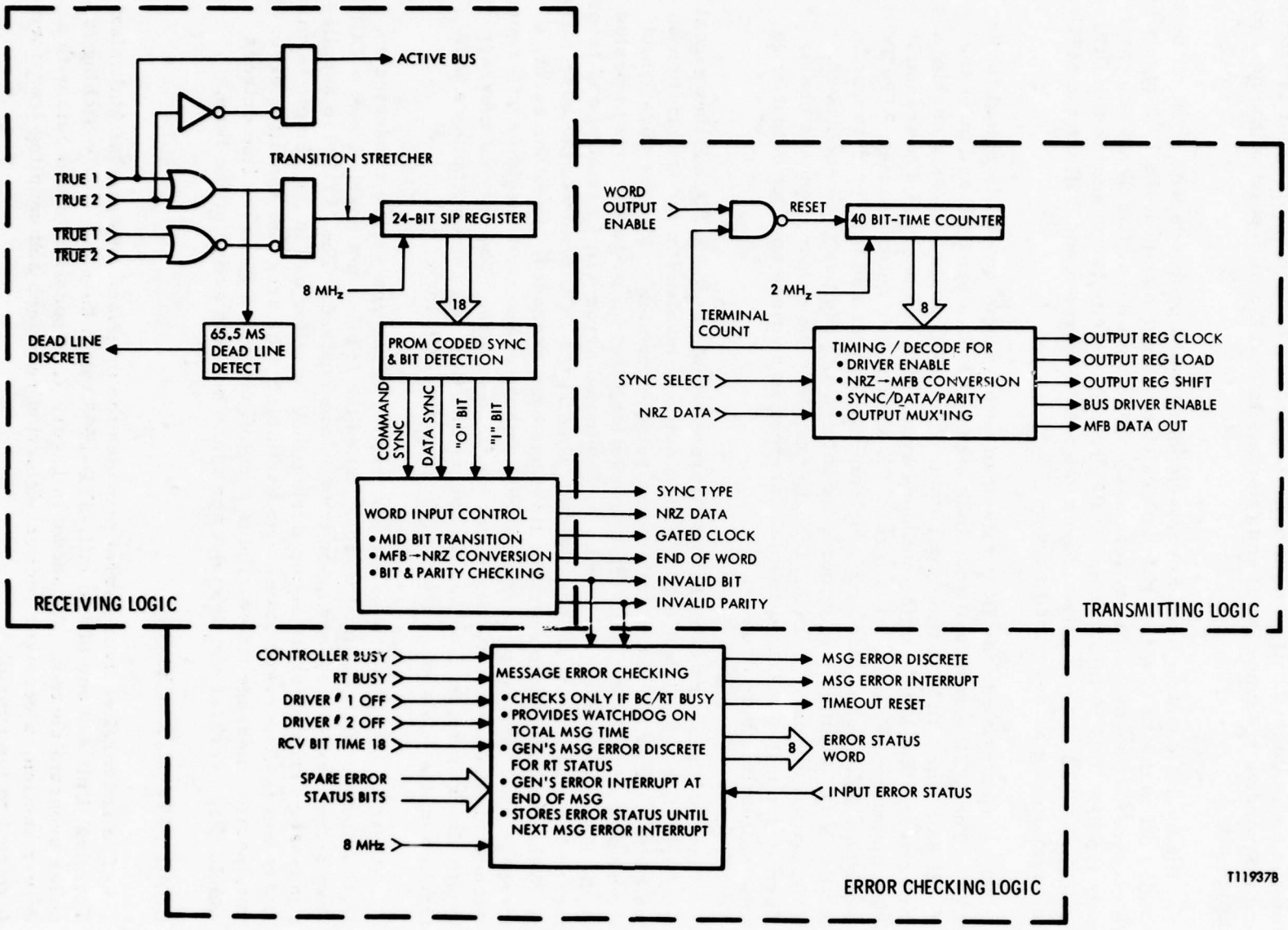



Figure 4. 1553 MTU Block Diagram

and the bit is latched. If no data pattern detection occurs, a bit error is latched and the window re-syncs itself and attempts to find the mid-point of the next predicted bit.

Bit-by-bit construction of every incoming word proceeds with each bit being handed off to the DCU with a gated clock (1 clock per bit) until the 17th parity bit is received. Should the parity bit not compare with the accumulated odd parity over the previous 16 bits, an error flag is set. Parity is not sent to the DCU, however an end of word pulse is transmitted. Figures 5 and 6 illustrate MTU receiver timing as described above.

To output a word, the DCU raises an output sequence enable signal to the MTU. The MTU sends load and clock signals back to the DCU and pulls the 16-bit data word bit at a time for output. The MTU posts a master enable to both drivers that brackets the outbound sync, data and parity. The transmit logic operates at 2 MHz, creating a TTL Manchester output format. The DCU supplies a sync select line, high for command words and low for data words, and the MTU supplies the proper sync pattern and odd parity to each word. While outputting parity, the MTU logic examines the output sequence enable discrete from the DCU to determine whether to continue another output or to go idle disabling both drivers.

The DCU supplies two "busy" discrete signals to the MTU. Either signal in a high state signifies that the DCU is actively involved in the transaction on the bus as either a bus controller or a remote terminal. For the duration of either busy signal, the MTU logic checks for valid bit or parity errors posted at the end of each received word. Any detected error will be latched and transferred to a program readable register when 1) the DCU removes the busy signal, or 2) the total allowable message time has been exceeded. In either event, a message error interrupt is issued and contents of the error register will remain latched until replaced by subsequent message errors. The current message length timer is set for 768 $\mu$s. Reaching this time out period implies a malfunction on the bus and a reset is issued to the DCU logic.

Because the MTU senses all bus traffic, error checks are conducted on DCU output as well as input words. When the MTU is not enabled with a DCU busy signal, errors occurring on the bus are ignored. The MTU also supplies a message error discrete signal during DCU busy for use in the remote terminal status word. The error status byte available to the program specifies error type (bit, parity, message timeout) plus shut-off discretes from the line drivers should they overheat and drop out and which bus was active at the time.

## DUAL DRIVER/RECEIVER

The Driver/Receiver hybrid provides the interface between the Multiplex Terminal Unit and one of two MIL-STD-1553A data busses. Bus switching is under program control. As shown in Figure 7, it consists of two sets with a driver section, a receiver section, isolation resistors and coupling transformers (external to the hybrid).

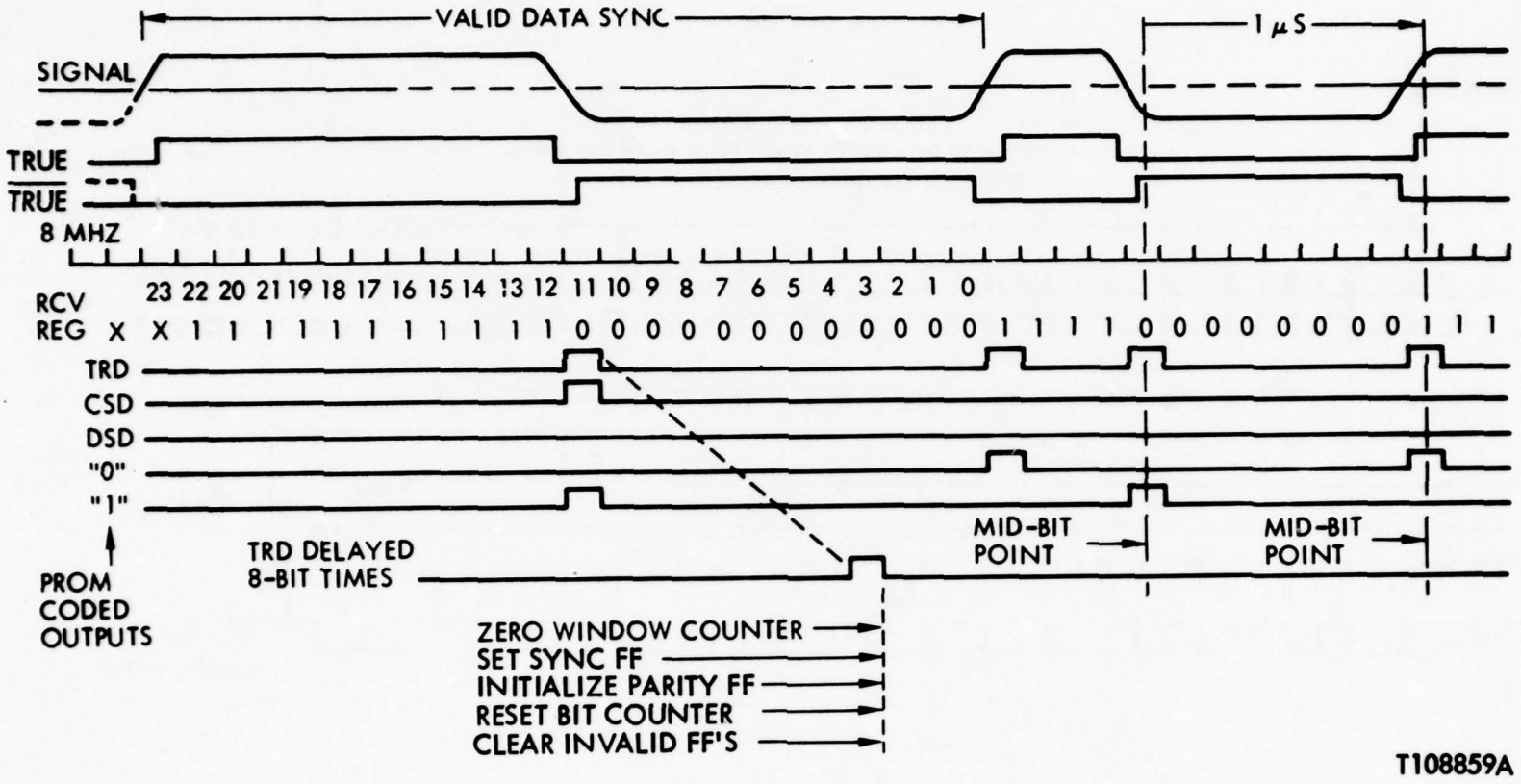



Figure 5. MTU Sync Detection, Bit Synchronization, Bit Validation Timing

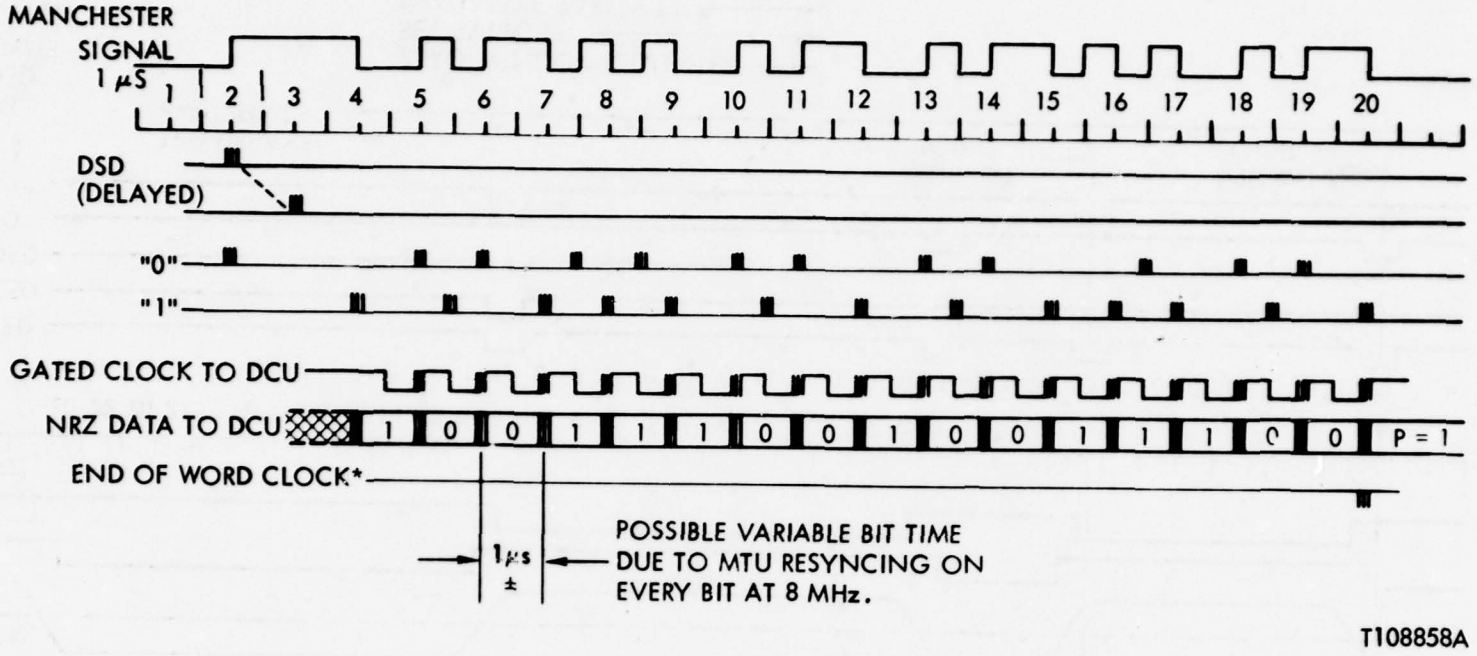



Figure 6. MTU Input Word Timing

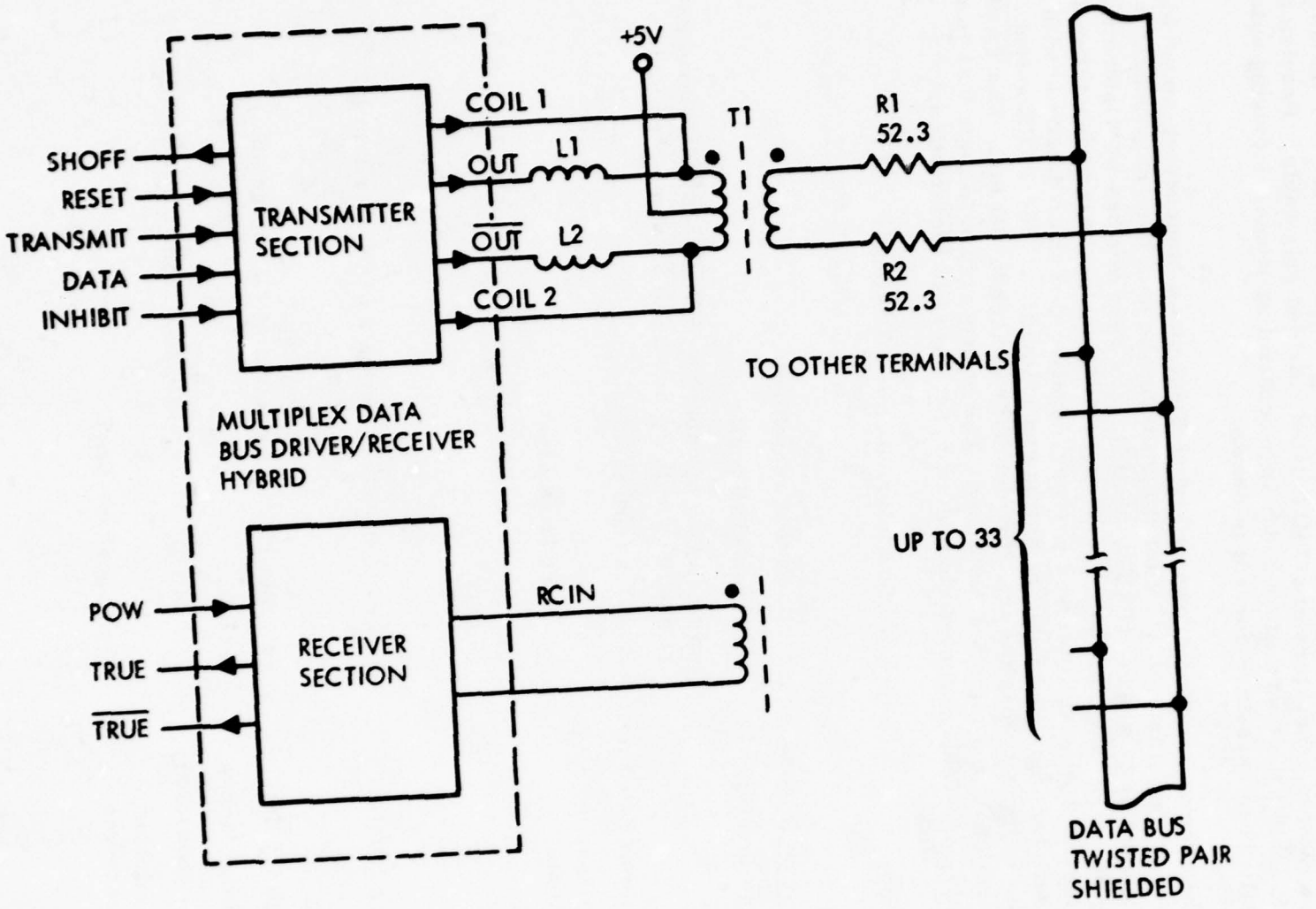



Figure 7. Bus Coupler Block Diagram

The driver section shown in Figure 8 is designed to accept a serial TTL data stream and produce a bi-phase signal of ±3V minimum to ±10V maximum between the two conductors of a 70Ω twisted shielded pair cable. Provision is made to inhibit the transmitter under logic control or it ceases driving automatically when excessive current is drawn.

The receiver section shown in Figure 9 accepts a Manchester-coded bi-phase signal of ±2V minimum to ±10V maximum from the data bus and produces a pair of TTL compatible signals (TRUE, $\overline{\text{TRUE}}$). The receiver has provisions for turning power off when not in use. The receiver design features an active four pole gaussian low pass filter and positive/negative threshold detection with hysteresis for improved noise rejection and reliability. The TRUE output switches to logic 1 when the transformer output exceeds 600 mv. The $\overline{\text{TRUE}}$ output switches to a logic 1 at -600 mv. The outpus return to logic 0 at ±6 mv, respectively. Table 2 summarizes driver/receiver characteristics.

Table 2. Bus Interface Specifications

| | |
|---|---|
| **Transmitting** (measured after isolation resistors) | |
| Voltage levels: | ±3V min ±10V max |
| Load: | up to 300 ft. of cable ($Z_o$ = 70Ω ±10% line terminated at each end with $Z_o$ and up to 33 other terminals |
| Rise time/Fall time: | ≥100 ns (10% to 90%) |
| Zero crossing deviation: | ±25 nsec from ideal |
| Protection: | shorts to ground and current exceeding 360 ma for 30 μsec |
| **Standby** | |
| Voltage levels: | return to 0 ±50 mv within 2 μsec after last bit |
| **Receiving** | |
| Voltage levels: | ±2V min ±10V max (clips at 2V, selectable) |
| Input impedance: | ≥2200 |
| Noise conditions: | S/N 14 DB, white, gaussian from 1 kHz to MHz |
| Noise emitted from receiver: | 10 mv peak-to-peak |

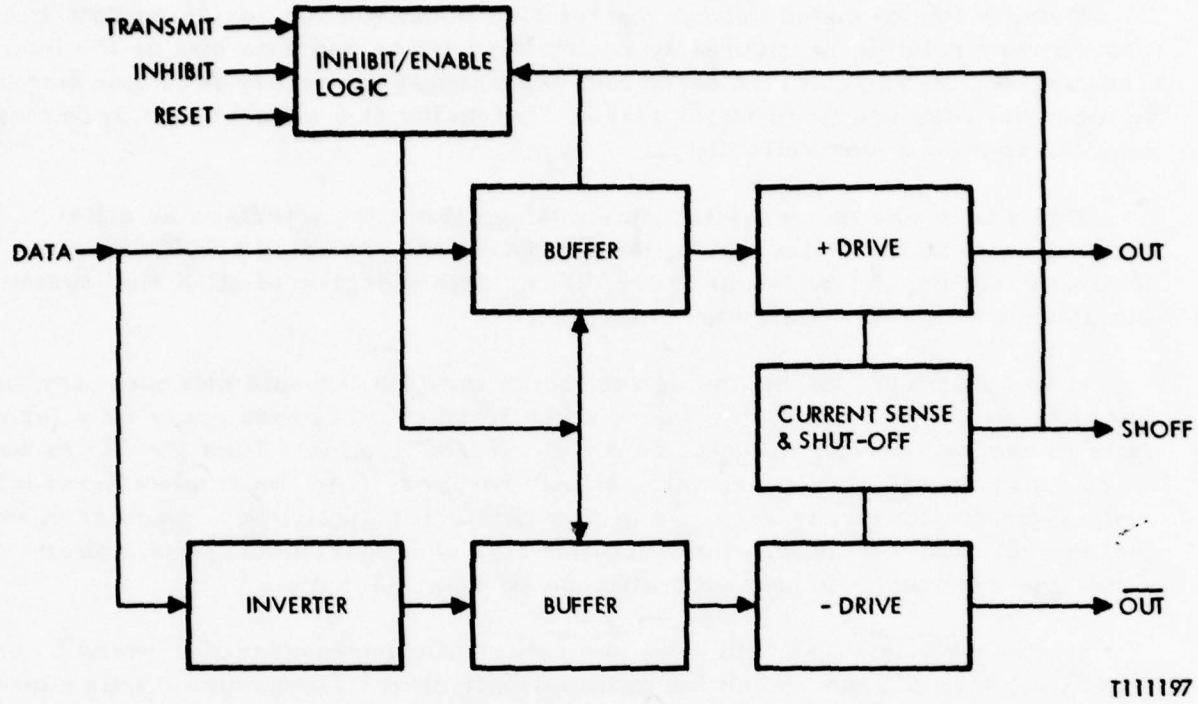



Figure 8. Driver Section

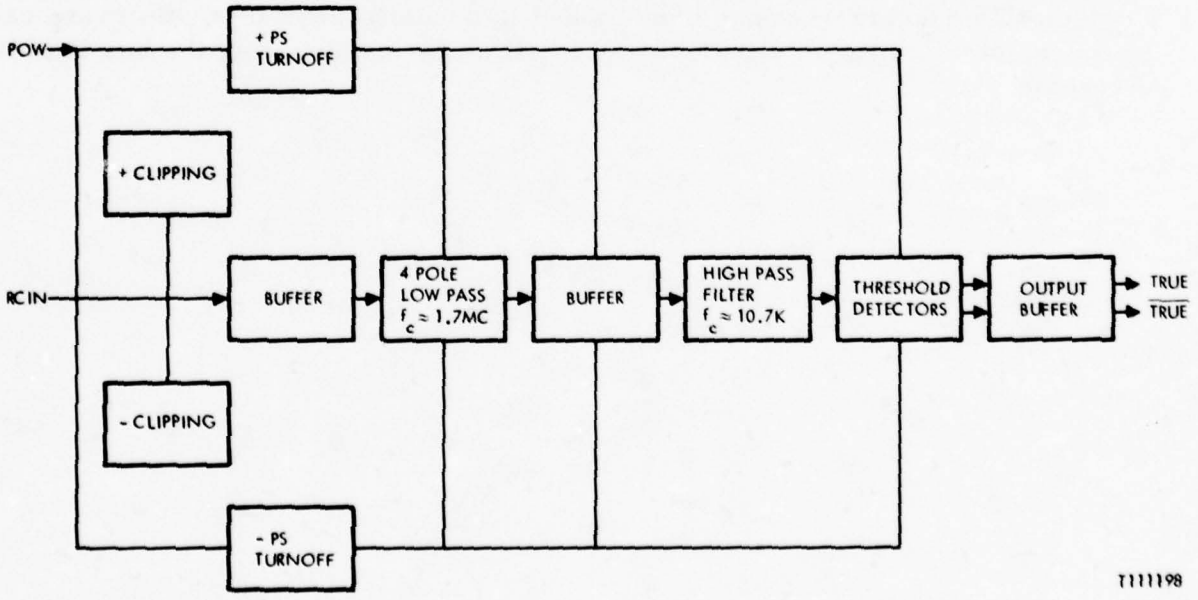



Figure 9. Receiver Section

SUMMARY

Using a PROM coded pattern correlation technique for receiving data has proved very reliable in accurately recognizing syncs and data bits on the bus. The stored patterns can be altered and expanded as necessary for noise immunity to meet the required word error rates. Resyncing at 8 MHz on each incoming data bit also enhances reliability.

Being able to receive while transmitting allows the interface as a Bus Controller to address itself as a Remote Terminal providing a 100% closed-loop self test capability. It also permits a certain degree of MUX Bus system simulation with only 1 hardware interface.

It was felt from the beginning that some time limit would be necessary on every message – in particular where short word count errors occur or a terminal fails to respond leaving a bus tied up with no traffic on it. Thus the 768 μs hardware timer was implemented which allows recovery from incomplete transactions. A dead-line monitor was required in this particular application to assist in switching control from the prime bus controller to a secondary backup controller. This dead-line discrete indicates no traffic for at least 65.5 ms.

It was considered best to have computer software manage the interface on a per-message basis when acting as Bus Controller. The program thus knows when and where errors occur and may alter its course of action as necessary. Minimum demands are made on software time. The program can output controls for the next message while the current message is in progress. A message in progress will not stop under program control. It will go to completion and immediately execute what has been loaded in queue (if anything). Software can, however, stop traffic on either bus at any time by manipulating the bus inhibit discretes.

# TOWARDS A UNIVERSAL SYNCHRONOUS COMMUNICATIONS CHANNEL UTILIZING MICROPROGRAMMED DEVICES

L. L. King
Westinghouse Defense and Electronic Systems Center
Baltimore, Maryland 21203

## Abstract

This paper evaluates the feasibility of utilizing microprogrammed devices in establishing a universal digital communications channel. A modular approach is utilized to achieve higher performance in line speed(s) and/or the handling of complex protocols by increasing the number of microprocessors along with other LSI devices utilized. Communications functions are distributed to the microprocessors associated with a specific configuration without significant destruction of the utility of the microcode postulated for a lower performance configuration.

## 1.0 Introduction

Universal is defined by Webster as "including or covering all or a whole collectively or distributively without limitation or exception." As applied to a communications channel or any other item, set, or function, this idea of universality is not in actuality or practicality achievable. However, with the advent of communications standards, medium (MSI) and large-scale integration (LSI) and especially microprogrammed devices, many barriers and constraints which existed have been diminished or removed.

## 2.0 Scope

This paper will investigate the feasibility of developing a "universal" communications channel utilizing microprogrammed devices. The approach considered is to begin with a small set of microprogrammable devices which provides a basic communications channel capability and utilizing modular techniques for both firmware and hardware, increase the communications channel capability to a high throughput device. This consideration is limited to the necessary functional processes, the functional process allocation to different microprocessors as more complex channel configurations are considered, and the probable capability achieved by each channel configuration considered. Finally, since asynchronous

techniques are so unique and are limited in line speed operation, this paper will consider, only, synchronous type information transmission.

## 3.0 Functional Requirements

Several functional processes are required to achieve effective operation of a digital communications channel when LSI/MSI microprogrammable devices are utilized or are required to operate a serial digital communications device itself. These include but are not limited to the following:

a. Serial Line Interface Characteristics
b. Microprogrammed device characteristics
c. Line and/or Channel Protocols
d. Computer/Channel Interface Characteristics
e. Support Logic Requirements

### 3.1 Serial Line Interface Characteristics

Serial Line Interface Characteristics defines the electrical parameters of the transmission medium, line drivers and receivers, and, if necessary, modem controls utilized for transmission of an electrical signal between two points.

#### 3.1.1 Transmission Medium

This applies, only, if the transmission medium is utilized to directly couple or "hard-wire" two or more communications channels together. The most popular mediums utilized are twisted pair or coax or triax cabling.

#### 3.1.2 Line Drivers and Receivers

The line drivers and receivers are defined in several "commercial" standards provided by the Electronic Industries Association. This includes EIA RS-232, RS-422, RS-432, and RS-439. Military requirements similar to the EIA specifications are described in the appropriate military specifications and standards.

#### 3.1.3 Modem Control

Modem interfacing and control is described in the appropriate commercial and military specification.

### 3.2 Microprogrammed Devices Characteristics

The performance levels of these LSI and MSI devices are controlled by the device technology utilized (TTL, ECL,

MECL, etc.), the architecture of each device, and capabilities provided by the devices for construction of larger systems when multiple devices are utilized.

Two categories of these devices are available for utilization. These include the "controller-on-a-chip" and the "bit slice" elements. The controller-on-a-chip device provides all of the logical operations necessary for functioning on one chip. The bit slice technology provides a certain number of parallel bite devices and requires supporting devices to satisfy the necessary logical operations requirements. The "bit/slice" technology provides higher logic rates than "device-on-chip" technology. However, some of the new single chip controllers are approaching the bit/slice technology capability.

### 3.3 Line and/or Channel Protocols

Protocols are defined as sets of procedures required to cause transmission and/or receipt of data across the communication line. The protocols define the synchronization, data and timing scheme(s) utilized for alignment and movement of data across the transmission medium.

Two different categories of protocol are the most widely used. These include character or bit oriented protocols and are described in the appropriate commercial or military specification.

### 3.4 Computer/Channel Interface Characteristics

The basic task of any communications channel is the transfer of data between two or more computers and/or user devices. To accomplish this task the communication channel must be tied to or interfaced with the higher order devices. The modern data channel must assume a major role in the operations of interface. These functions include the logic for data transfer usually as a direct memory (DMA) or programmed I/O operation responding to functional control from the higher order device and performing the hand-shaking required.

### 3.5 Support Logic Requirements

These are defined as those functions required to support logical operation of the channel. In other words, this logic does not contribute to the logical function for handling or processing the information for message flow or control but actually perform support task(s) for logic performing the message flow.

## 4.0 Channel Operation Evaluation

The primary objective of this paper is to evaluate the probability of developing a universal synchronous digital communication channel using microprogrammed devices. A series of functional block diagrams is utilized to demonstrate the modularity of these devices and to show the allocations of functions between the microprocessor(s) shown in the different block diagrams.

Capability estimates for the different configurations are based on experience gained by Westinghouse in the integration of complex computer systems and detailed studies performed for several different customers. These capability estimates are considered to be conservative.

### 4.1 Basic Performance Level

Figure 1 shows the basic or simplest level micro-device system capable of performing a communications channel function. Note that only one microdevice is utilized to perform all the functional task necessary for communication channel operation. One microprogram provides complete channel support of the functions listed on the diagram.

### 4.2 Improved Performance Level

Figure 2 shows an improved performance level system capable of performing a complex communications channel function. This improved performance includes the capability of handling a higher level protocol, DMA computer/channel operation and higher line speeds. This is achieved by adding a second microprocessor to the basic system shown in Figure 1 and distributing the functions between the two microprocessors. The distribution of these functions are listed in Figure 2.

Note that one additional functional parameter had to be added to both processors. This is the communciations required for each micro-device to synchronize with each other.

### 4.3 Higher Performance Level

Figure 3 shows a system capable of performing the higher level protocols and increased throughput speed when compared to the system shown in Figure 2. This is achieved by adding a RAM buffer and a LSI protocol processor to the Figure 2 configuration.

The LSI protocol processor can either be a LSI device that is specifically designed for the selected protocol or a

microprocessor specifically programmed to handle the selected protocol. In either case, the specially designed device is much more efficient than the programmed microprocessor.

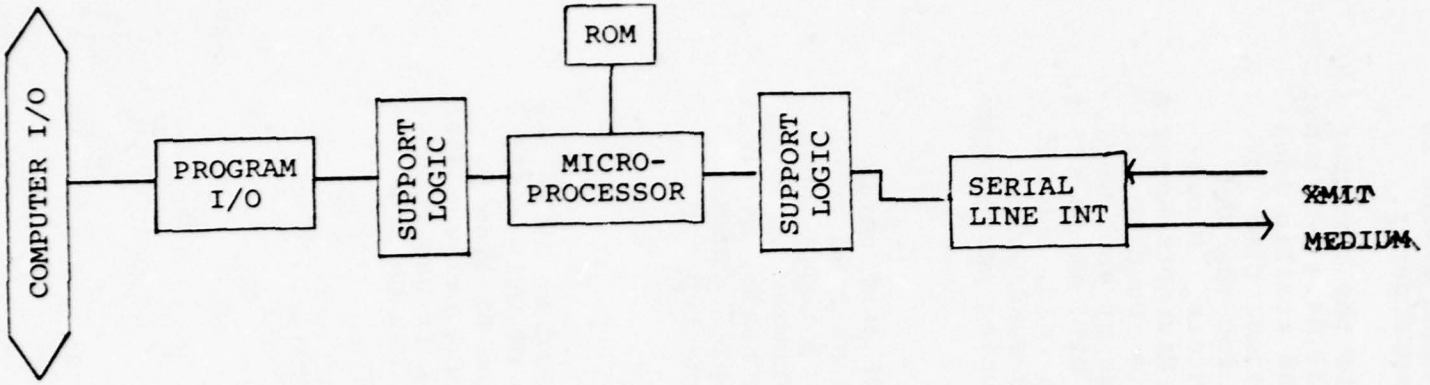
The RAM memory is utilized as a buffer to the computer I/O system and eliminates the requirement that all data be transferred through the microprocessors. This reduces the traffic load on the microprocessors and permits more processor time for performing other functions. Address control for the RAM memory is performed by using two address pointers. Microprocessor B controls the "Write" pointer and Microprocessor A controls the "Read" pointer. One extra bit is added to all memory addresses to show the read/write status of each word. When the memory location is written into the bit, it is set to a logical "one". When the memory location is read, the bit is set to a logical "zero". By checking this read/write status the microprocessors assure non-overlapping performance.

## 4.4 Maximum Performance Level

Figure 4 shows a configuration capable of performing the communciations channel function at a near maximum rate. Note that these are duplicate configurations of those shown in Figure 3 except each configuration is programmed to perform as a simplex channel. Two simplex channels, one for input and one for output, perform the full duplex channel tasks. The two simplex channels are interconnected for coordination of functionally assigned tasks.

## 5.0 Conclusions

By using a modular approach in the development of a digital communications channel a large degree of universality can be achieved. It was shown that a wide band of line speeds can be achieved (1.2K - 1M b/s). The channel can also be expanded to handle higher level protocols. It was further demonstrated that channel operation functions can be distributed between microprocessors.



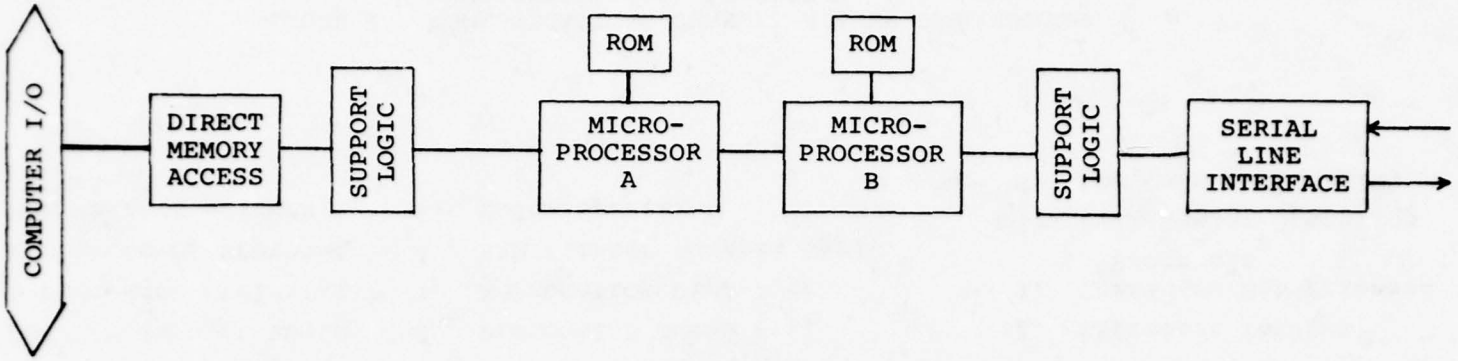

MICROPROCESSOR FUNCTIONS

1. Line Interface
2. Computer I/O
3. Protocol
4. Support Logic
5. Modem Control

CAPABILITY

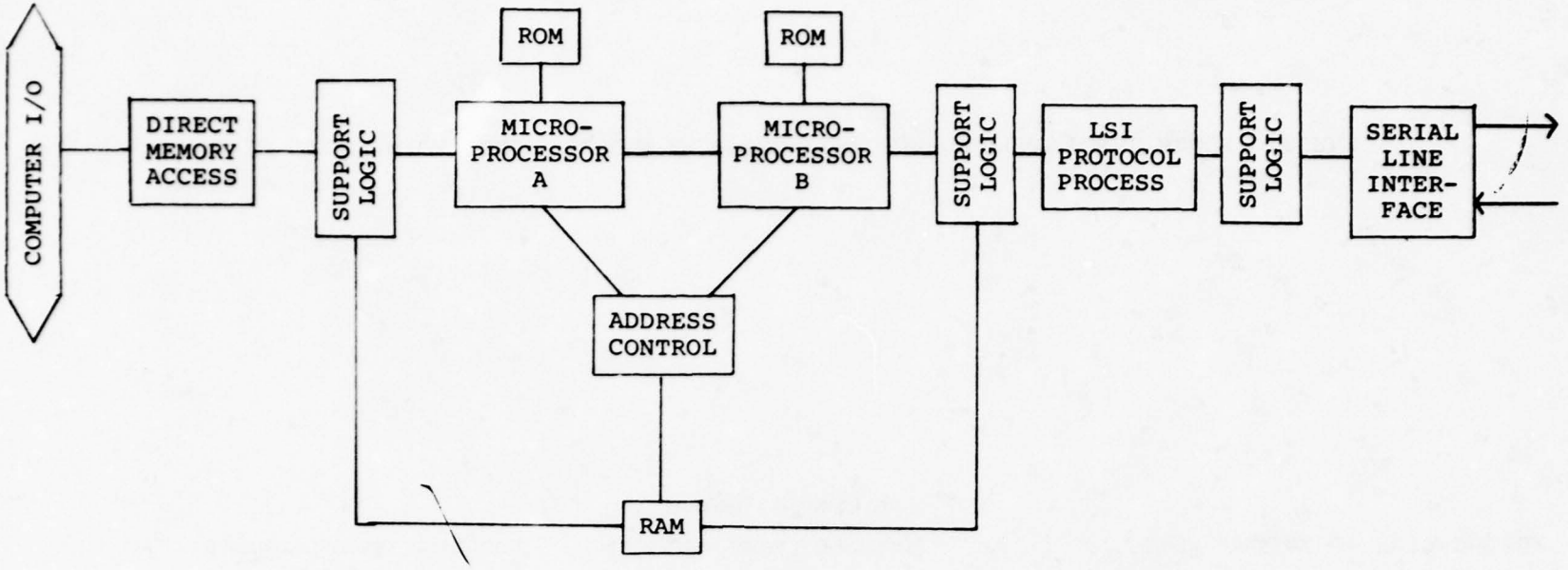
1. Line Speed ($\leq$ 9.6Kb/s)
2. Throughput Inefficient
3. Simple Protocols
4. Simple or Half Duplex

FIGURE 1 COMMUNICATION CHANNEL BASIC PERFORMANCE FUNCTIONAL BLOCK DIAGRAM




| MICROPROCESSOR FUNCTIONS | | CAPABILITY |
|---|---|---|
| A. | B. | |
| 1. Direct Memory Access | 1. Serial Line Interface | 1. Line Speed (≤100K b/s) |
| 2. DMA Support Logic | 2. SLI Support Logic | 2. Effective Thruput |
| 3. B Processor Xfer | 3. A Processor Xfer | 3. Character Oriented Protocol |
| 4. Higher Level Protocol | 4. Low Level Protocol | 4. Half Duplex or Full Duplex |
| | 5. Modem Operation | |

FIGURE 2 COMMUNICATIONS CHANNEL IMPROVED PERFORMANCE FUNCTIONAL BLOCK DIAGRAM




648

MICROPROCESSOR FUNCTIONS

A.
1. Direct Memory Access
2. DMA/RAM Support Logic
3. B Processor Ctrl Xfer
4. Higher Level Protocol
5. RAM Pointer Address Ctrl

B.
1. Serial Line Inter Ctrl.
2. Protocol Process Ctrl.
3. A Processor Ctrl Xfer
4. RAM Pointer Address Ctrl.
5. Modem Operation

CAPABILITY

1. Line Speed ($\leq$500K b/s)
2. Efficient Thruput
3. Character/Bit Oriented Protocols
4. Full Duplex Operation
5. Low Computer Loading

FIGURE 3 COMMUNICATIONS CHANNEL HIGHER PERFORMANCE FUNCTIONAL BLOCK DIAGRAM

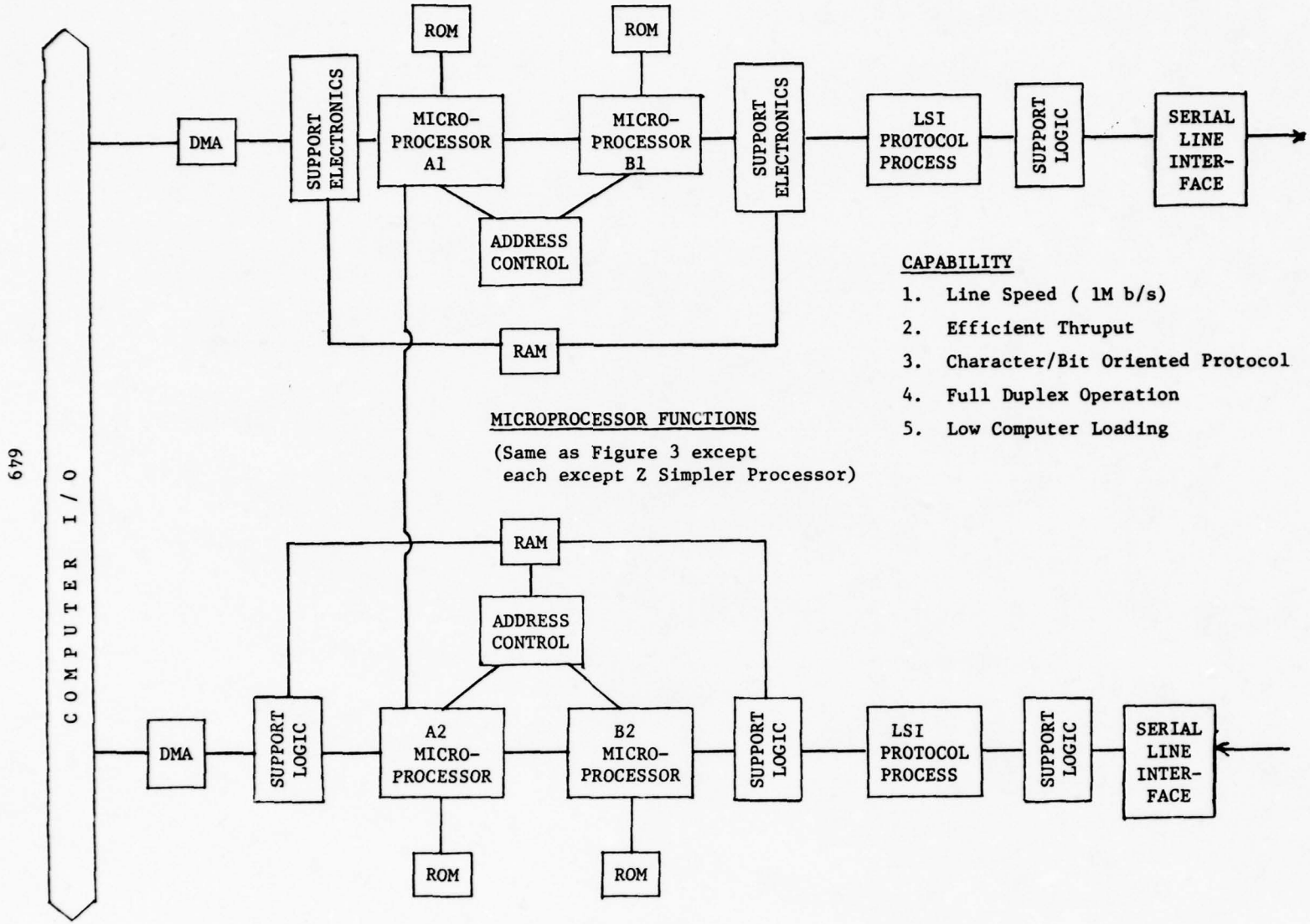



FIGURE 4 COMMUNICATIONS CHANNEL MAXIMUM PERFORMANCE

MULTIPLEX DATA BUS CONFERENCE

10-12 OCTOBER 1978

DAYTON, OHIO

SESSION VI

# TESTING AND EVALUATION OF MULTIPLEX SYSTEMS

MODERATOR: Colonel George L. Von Tersch
ASD/ENA
W-PAFB, OH

PRECEDING PAGE BLANK

# 1553 BUS TESTERS

Nancy L. Esken and Anthony L. Haley

ASD/ENASC, Wright-Patterson AFB, Ohio

## ABSTRACT

The Systems Engineering Avionics Facility (SEAFAC) contains a MIL-STD-1553 Data Bus Test Bench. The Data Bus has various remote terminals connected to it and it was determined that some type of test equipment was needed in order to initially checkout hardware connected to the data bus. Since that time several pieces of equipment have been built. Each of them is called a Bus Tester and is numbered consecutively in the order of design (i.e., Bus Tester I, Bus Tester II, etc.). This paper will describe the reasons for and general operations of each of three generations of Bus Testers.

## INTRODUCTION

There are several reasons for a Bus Tester - one of which is to keep the hardware and software as far apart as possible. Without some type of Bus Tester, when a hardware device has been built, the hardware person has to rely on software personnel for checkout. This can lead to headaches determining whose wrong, software or hardware. By using the Bus Tester as a BUS CONTROLLER, initial hardware checkout can be accomplished without the need for any software.

A Bus Tester is also needed for monitoring the Bus. Before when a person looked at the Bus with a scope, with the BUS CONTROLLER talking asynchronously to more than one device, it is difficult to sync on a particular message, and even more difficult to read the manchester biphase on the scope. Using a Bus Tester which can selectively pick



PRECEDING PAGE BLANK

off messages and display them allows a person to look at Bus Traffic for one RT. Another reason for a Bus Tester is for error detection and error flagging.

OPERATION OF THE BUS TESTERS

The first generation Bus Testers are fundamentally limited in their capabilities and are used mainly in initial hardware checkout and periodic maintenance checks. They perform two functions; that of a bus monitor and an elementary bus controller. A switch on the front panel determines the mode of operation.

When the Bus Tester acts as a bus monitor, it displays all traffic presently on the data bus. Displayed on the front panel are the most recent command word, the most recent status word and the two most recent data words transmitted on the data bus.

When the Bus Tester acts as a bus controller, it sends commands to a remote terminal and communicates with this remote terminal. The Bus Tester outputs a sixteen bit command word, entered through the front panel switches. The sixth most significant bit is the transmit/receive (T/R) bit. If T/R is high, the remote terminal is told to transmit a status word and the specified number of data words to the Bus Tester, which displays them on the front panel. If T/R is low, the remote terminal is told to receive data words from the Bus Tester that are entered through the sixteen data word switches on the front panel. The remote terminal responds with a status word, that is displayed by the Bus Tester.

Bus Tester II was the follow-on, or second generation Bus Tester designed and built within SEAFAC. Only one module unit has been constructed. This tester provides all Bus Tester I capabilities plus an expanded capability to provide pre-programmed monitoring of the 1553A data bus and pre-programmed message transfer on the bus. It also provides a means to display various error rates. This gives a sophisticated means of monitoring remote terminals for initial checkout, maintenance and bus observation. It has more extensive bus controller capabilities than Bus Tester I. This tester has been used for validation of some contractor-developed equipment and has been requested for future validation checks by others. The design documentation has been requested by two companies and the legal

implications of providing such have been cleared through the base Judge Advocate General office. The design has been furnished to Northrop Corporation, Electro-Mechanical Division for use in their test facility, and it will shortly be sent to the Hoffman Electronics Company.

Operation of the Bus Tester is via front panel switches. A PROG/XEQ switch specifies whether the tester is in the programmed phase or the execution phase. In the program phase, command words and data words are entered using the front panel switches. For example, to enter a command word the 16 switches under the displays are set and the ENTER COMMAND switch is pressed. Data words are entered in the same manner using the ENTER DATA switch in conjunction with the Data Address control switches. After programming the tester the PROG/XEQ switch is placed in the XEQ position and the initiate push button is pressed. The tester has two modes of operation which are controlled using the front panel switch XMIT/LISTEN. In the TRANSMIT mode the tester acts as a bus controller and transmits a pre-programmed command word and data words if it is a receive command at a rate specified by the XMIT Delay switches. In the LISTEN mode the tester monitors the bus and displays messages corresponding to a pre-programmed remote terminal address. The tester can also detect certain errors such as a status no response from a particular RT, low or high word counts and if the T/F or M/E bit is set in a particular RT status word. Control of error detection is done using front panel switches.

A third generation Bus Tester was needed to further test the operation of remote terminals connected to a data bus. Terminals which operate properly when sent good information may not necessarily operate in a desired manner when sent bad information. Bus Tester III will have all the functions of Bus Tester II plus the capability to do full RT verification and validation checks. This tester will provide fault detection tests for MTUs by generating errors such as bad sync for command word or data word, bad biphase, or bad parity. Bus Tester III will be rack mounted within a permanent test console along with a PROM programmer, Hazeltine 2000, Tektronix 611 CRT, two force stick hand controllers, and power supplies.

Operation of Bus Tester III is as specified in the paragraph above which describes the operation of Bus Tester II. A few minor changes have been incorporated in the

design and are readily discovered when the front panel layouts of the two testers are compared. When operating in the transmit mode (or as a controller) the tester can be programmed via the front panel switches to transmit the following types of errors: parity, short or long words, too few or too many data words, encoding errors, variable data rate, and glitches on the transmission line. The following is a brief description of each:

- parity - any single word may have a parity error or all words can have their parity bit either held high (to a logic one), held low (to a logic zero) or inverted.

- short or long words - all words transmitted by the tester can be expanded or reduced by a maximum of three data bits.

- too few or too many data words - any number of data words different than the number specified in the transmitted command word.

- encoding errors - changing the biphase code and sync of any word transmitted.

- variable data rate - changing the bit transmission rate by 1 MHz $\pm \Delta f$.

- glitch on line - inflecting narrow pulses anytime during or between transmissions.

In addition to the error infliction capability, Bus Tester III can operate as a controller for remote terminal-to-remote terminal transfers. The tester also has outputs for single-line hookup to a terminal without the use of a main bus line and coupler boxes.

# MIL-STD-1553 MONITOR INTERFACE FOR FLIGHT TEST

James L. Voight
Donald H. Ellis

SCI Systems, Inc.
Huntsville, Alabama

## ABSTRACT

This paper describes a small, highly flexible interface unit designed for the exchange of selected information from a MIL-STD-1553 data bus to an advanced flight test instrumentation system. The unit can be used to monitor the on-board data bus for proper operation and utilize the bus data for aircraft performance verification.

The flight test instrumentation system employs an internal MIL-STD-1553 type high speed serial digital data bus for remote data acquisition. Therefore, the unit serves as a monitor at the aircraft bus and a remote terminal for the instrumentation bus, providing a buffered interface for the asynchronous data busses.

The bus monitor is designed for compatibility with all known MIL-STD-1553, Revision A and B bus interfaces and protocols. The unit control logic is microprogrammable for flexibility in case future modifications of bus protocol or timing are required. The decom program resides in an alterable non-volatile memory providing flexibility for aircraft bus sampling program changes. The design and construction allows for easy adaptation of the instrumentation interface to other system applications requiring interface with MIL-STD-1553 data busses.

## ADVANCED AIRCRAFT FLIGHT TEST SYSTEM

The advanced aircraft flight test instrumentation system is designed as flexible general purpose equipment for the Air Force Flight Test Center, Edwards AFB, California, and is commonly known as Airborne Test Instrumentation System (ATIS). ATIS has been developed to provide a standard set of instrumentation hardware for current and future Air Force flight test programs and is in use on AWACS, Command Post and F-16 Aircraft.

ATIS is a flexible modular data acquisition system which employs stored program remote multiplexing over an internal serial digital data bus (party line). .Figure 1 shows an ATIS configuration with major hardware elements

and the monitor which interfaces the MIL-STD-1553 bus with the ATIS internal data bus. The basic ATIS system configuration consists of System Control Unit communicating with analog and digital remote units via a high speed serial digital data bus (party line). The remote units are used to sample a large number and wide variety of airframe and engine measurements throughout the aircraft. The ATIS hardware is pictured in Figure 2 photograph. The following is a brief description of the basic ATIS system hardware components and the instrumentation data bus.

System Control Unit - Provides control of up to 32 remote multiplexer units, analog, digital, or mix, from a preprogrammed non-volatile memory. The SCU accesses the selected remote units and channels with address/control words and formats the received data for output to real time telemetry or on board tape recorder. Storage is provided for up to 16 software controlled sampling programs and data formats with 8 selectable output PCM bit rates from 4kbps to 512kbps.

Analog Remote Unit - The ARU operates under control of the SCU via the high speed party line and provides 30 channels of analog multiplexing. The analog input signals may range from ±20MV to ±10 volts full scale. Preprogrammed gain control is provided for each channel from the SCU with 8 selectable gains from 1 to 500. The analog samples are converted to 11 bits plus sign digital words in a high accuracy self-calibrating A/D converter. The 12-bits are transmitted to the SCU via the party line for processing and formatting.

Signal Conditioning Assembly - The programmable gain and offset features of the ATIS allow the ARU to handle the output of most transducers directly without additional signal conditioning. The Signal Conditioning Assembly (SCA) provides flexible modular signal conditioning, filtering, transducer excitation and remote voltage sensing at the transducer terminals for situations requiring maximum accuracy. The SCA accommodates up to ten (10) signal conditioning modules of nine (9) types.

Submultiplexer Unit - Analog channel capacity of the ARU can be expanded by the addition of the submultiplexer. Each submux allows up to four (4) channels of the ARU to be expanded by 128 high or low level inputs, 32 for each ARU channel.

Digital Remote Unit - The Digital Remote Unit (DRU) is a modular assembly for the conditioning and acquisition of digital data. The DRU consists of a Digital Remote Interface Unit (DRIU) and up to eight (8) modules which provide the conditioning for six (6) signal types, including serial and parallel discretes, frequency, elapsed time counter, tape header generator, and pulse totalizer. An additional module for decoding high power output discretes is provided for driving lamps, relays and stepping switches.

General Purpose Airborne Processor - The GPAP subsystem hardware is available for optional expansion of ATIS capabilities to include real time in-flight data processing; display and printing. The subsystem contains an Airborne Digital Computer, Digital Line Printer, Pilots Panel Display and control switches for program control and print routine selection. Selectable software programs provide data reduction, data compression and any other processing routine that can be programmed in the available memory.

Optional System Controller (Format Control Unit) - The Format Control Unit (FCU) is a low cost unit which provides system control for applications not requiring the full capability of the ATIS system. The FCU utilizes a solid state EPROM for program storage, has a capacity of 8 remote units and provides a maximum PCM output rate of 128kbps.

The FCU Programming System provides an easy method of memory program preparation, checkout using a RAM memory, and loading and verification of the EPROM program.

Ground Support Unit - The Ground Support Unit (GSU) provides a capability for operation and checkout of the ATIS in laboratory or flight line environment. The major components of the support system are a ruggedized minicomputer, interactive display keyboard, cartrifile and a control and format synchronization subsystem. Multiple functions are performed under control of "prompting" software which provides maximum operator capability with minimum training required. A photograph of a typical mobile support unit for pre-flight checkout of the system on the aircraft is shown in Figure 3 photograph.

Instrumentation Data Bus - The ATIS data bus (party line) is full duplex utilizing two twisted shielded pair cables operating at 1.5mbps. One cable is the address/command line from the System Control Unit (SCU) to the remote multiplexers carrying remote unit address, channel address and gain setting for the analog channels. The other cable carries the data return from the remote units to the SCU. The modulation technique is biphase manchester, with 20 bit words and the twisted pair is transformer coupled at the terminals in the same way as MIL-STD-1553. The word and message structures are different from 1553 due to the unique requirements of the instrumentation system. The bus was designed prior to the finalization of the MIL-STD-1553 original version. More details of the instrumentation bus will be covered in the data bus monitor interface description.

Data Bus Interface Unit - Data Bus Interface Unit (DBIU) is the designation chosen for the ATIS MIL-STD-1553 monitor/decom interface unit. The

DBIU functions as a remote unit on the instrumentation data bus. MIL-STD-1553 transmissions are decoded and selected words pre-stored in the DBIU data buffer memory. The preprogrammed memory locations are addressed and selected data is acquired by the ATIS control unit via the high speed instrumentation data bus. A detailed description of the DBIU design and functional capabilities are included in the sections to follow.

## FUNCTIONAL DESCRIPTION

The DBIU provides programmable decommutation capabilities for up to four (4) MIL-STD-1553 aircraft data busses, data memory storage for 255 data bus words or time-tag correlation words, and the interface with the ATIS instrumentation data bus. The alterable non-volatile decom memory is functionally organized as message list memory for aircraft data bus message identification, and word list memory for bus word and time-tag selection. The data memory supplies 16 data bits, the parity bit, bus identification bits, and data update rate flags for aircraft data bus words, and 20 bits of time-tag data to the instrumentation data bus. Control functions for the memories and bus interfaces are generated by a microprogrammable control sequencer. The block diagram for the DBIU is shown in Figure 4.

Avionics Bus Receiver - The bus receiver shown in Figure 5 interfaces four (4) aircraft data busses. The receiver input transformers contain sufficient windings and strap options to provide for aircraft data bus couplers utilizing either transformer or direct bus connections. With the exception of the input filters and threshold detectors, the design is totally digital, consisting of a rephasable 4 MHz clock synchronizer, pulse width discriminator, valid sync detector, error detectors for invalid Manchester biphase coding and word length, and registers for data, parity, bus I. D., and data status. The data status information provides sync type and validity, coding and word length error flags, and a data ready flag to the control sequencer logic for micro-program control.

The bus I. D. register supplies five I. D. lines to the data memory. Four of these lines indicate reception on busses 1 through 4 for ATIS system configurations in which the Control Unit is used in the 12 bit data reply mode. The fifth line distinguishes between data reception on bus 1 and 2, or bus 3 and 4, for the ATIS 10 bit data reply mode. A strap option is provided to convert this line to a bus 1/bus 2 flag for system configurations where only two aircraft data busses are monitored.

Decom Memory - The decom memory is implemented with three 2048 word by 8 bits/word EPROM's contained in a bolt-on module to simplify removal for decom format programming. The memory is organized for two functions, message identification and word selection within a message. The message identification function is contained in two EPROM's, each containing a separate

message list program selectable by either an external switch or a connector strap. The first eleven data bits of the aircraft data bus command word, the terminal address field, T/R bit, and the subaddress/mode field, supply the address to the message list memory. The memory data output provides a starting address to the address counter for the word list memory. An all 1's code from the message list memory indicates that no data is required from the current bus message, and is decoded and supplied as a flag bit to the control sequencer for microprogram control.

The word select function is contained in the third EPROM, which is organized as four 512 word programs selectable by either external switches or connector straps. Each word list program contains word select data for word type and data word count, and data memory storage addresses.

Data Memory - the data memory is a multiple-port, high speed bipolar memory organized as 256 words by 24 bits/word. The data memory word format is shown in Figure 6. Address inputs are received from the word list memory, time-tag interface, and the ATIS bus interface. Data inputs are received from the aircraft data bus receiver register and the time-tag interface. Data and address selection is made by the control sequencer. Memory address zero is dedicated to elapsed time counter (ETC) data and is updated at the ETC clock rate.

Aircraft data bus words are formatted as 16 data bits, the parity bit, 2 data update rate flags, and the bus I. D. flags. The update rate flags, the staleness (ST) and overflow (OV) bits, indicate the memory access rate differences between selected data from the aircraft bus and data requests from the instrumentation bus. The ST flag is set when 2 consecutive instrumentation bus requests are made for non-updated data from the aircraft bus. The OV flag is set when aircraft bus data has been stored twice between instrumentation bus requests.

Memory data for the instrumentation bus is transmitted in two reply words as shown in Figure 6. To ensure data correlation for the two reply words, the first request for a given address loads the 24 bit memory word into the instrumentation bus reply register. The second request transmits the second half of the word.

Time-Tag Interface - The DBIU provides two methods for time correlation, an internal elapsed time counter (ETC), and interfaces for external serial and parallel time sources. Decom selection of internal or external time-tags for data memory storage is derived from the word-list memory. The external serial/parallel time-tag source is selected by a remote switch or connector strap. The ETC is a 20 bit binary counter with selectable clock rates of 1 KHz or 10 KHz by a remote switch or connector strap. The counter

is reset by a microcode power-up initialization routine or an external reset input. Time-tag staleness between data word and time-tag memory storage is ±1 count.

ATIS Bus Interface - The ATIS instrumentation bus is full duplex, transformer coupled, 1.5 MHz, biphase Manchester, utilizing a 1553 type sync pattern. The address/command bus word structure for the DBIU utilizes 20 bits: 3 bit time positive sync followed by a biphase "One", 5 bit Remote Unit I. D., 8 bit data memory address, normal/sequential flag (ARU only), reply/no reply flag, and a final biphase "One". The unit I. D. from connector straps, the fixed data "Ones", and the reply flag are compared to the address/command word to generate data memory requests. The reply data bus utilizes 18 bit words: 3 bit time positive sync, biphase "One", 10 or 12 bit data, digital/analog data flag, and a final biphase "One". In the 10 bit reply mode, the two leading bits of the data field are zeroes. The digital/analog flag indicates to the ATIS Control Unit the option of addition/subtraction of preprogrammed offset codes for analog data.

Control Sequencer - Aircraft data bus message and word synchronization, and decom and data memory control functions are provided by a high speed, bipolar, microprogrammable sequencer. The sequencer assembly includes the control sequencer, microprogram memory, pipeline register, and a condition code (CC) mux. The control sequencer features 5-level subroutines, jump and branch instructions, and a loop timing counter. Instruction execution rate is 4 MHz. The microprogram memory is implemented with fusible-link, bipolar PROMS and configured as 512 words by 32 bits/word, while the architecture allows for up to 4096 words. Memory outputs are clocked into a pipeline register to provide stable data and reduce sequencer loop delays. The microcode data field includes sequencer instructions, CC mux input select, system clock, clear, and enable signals, and loop counter/jump and branch address values. The CC mux selects flag inputs for conditional jump and branch instructions. Inputs include aircraft data bus receiver status and command word T/R bit, decom flags from message list and word list memories, and connector strap flags for bus protocol selection.

## FLEXIBLE DECOMMUTATION/PROGRAMMING

The DBIU provides a number of decom memory programming features for aircraft data bus message and word selection, and simplified programming techniques. Memory word formats are shown in Figure 6. Message list data is contained in one byte, and word list data in two bytes. The word list starting address accesses programs contained in 512 word pages; so only the 8 MSB's of the address need be programmed. Two message list programs are externally selectable. Word list data contains two word formats, word type selection and data memory addresses. The word type byte contains an end-

of-message (EOM) bit to indicate that no further data is required from the current bus message, and word type and data word count fields. Four word list programs are externally selectable.

The word type field selects command, status, or data words from bus messages, and internal/external time-tag data for data memory storage. The word count field is utilized for bus data word selection, and internal/external time-tag selection by utilizing the word count LSB as a flag input to the control sequencer CC mux. Future bus protocol changes such as additional word types, word counts of greater than 32 words, etc., are easily implemented by reformatting the word list memory bytes/access.

## FUNCTIONAL MODULARITY

The DBIU electrical and mechanical design, and functional partitioning, provide simple, low cost adaptation to accommodate present and future unique protocols on both the aircraft and instrumentation data busses. The aircraft bus receiver interface is contained on two PC boards, instrumentation bus interface on two boards, the time-tag interface on one board, the data memory on one board, and the control sequencer assembly on one board. Possible adaptations include word length and bit rates on the serial busses, a parallel interface for the instrumentation bus, data memory expansion, and time-tag frequencies, word lengths, and external interfaces.

Along with the present flexibility provided by microprogramming and decom programming, the functional modularity of the DBIU provides highly flexible, adaptable bus monitor functions.

## PROGRAMMING/TEST EQUIPMENT

Test equipment is provided for flexible bench testing and decom memory programming without requiring an operational aircraft bus and ATIS system. A bus simulator provides the wide variety of bus messages, word types, error conditions and signal levels required to simulate the MIL-STD-1553 bus interface. Also included as part of the test equipment is an ATIS/DBIU Test Set. This unit provides a flexible DBIU control over a simulated ATIS instrumentation bus, and allows the functional capabilities of the DBIU to be tested while operating with a simulated or operational 1553 bus.

A microprocessor controlled Programmer is provided for flexible programming of the DBIU decom memory (EPROM). An internal RAM may be programmed manually or from an external support system and the program checked out with the DBIU prior to loading into the EPROM. The memory module is plugged into the Programmer front panel for erasure with a built-in ultraviolet (UV) light source. The program in RAM can then be loaded into the EPROM under control of the front panel switches.

## PHYSICAL AND ENVIRONMENTAL CHARACTERISTICS

As with all ATIS hardware the DBIU is designed for and qualified to the severe environmental conditions experienced in uncontrolled spaces on modern high performance aircraft. The following is a listing of the physical characteristics and operational environmental conditions which are a part of the DBIU design requirements.

- Size - 8.00 x 5.00 x 3.50 inches (basic housing similar to ARU, without decom memory module). 8.00 x 5.00 x 4.25 inches (with 8.00 x 5.00 x 0.75 inch decom memory module).
- Weight - 7.0 pounds, maximum
- Power Dissipation - 20 watts, max. at 28VDC
- Temperature - -54°C to +85°C
- Altitude - 0-70,000 feet
- Relative Humidity - 95% @ 71°C
- Vibration - 5 to 20 Hz @ 0.10 inch DA; 20 to 33 Hz @ 2 g accel.; 33 to 74 Hz @ 0.036 inch DA; 74 to 200 Hz @ 10 g accel.
- Electromagnetic Compatibility - MIL-STD-461A, CE03, CE04, RE04, RS04
- Shock - Half sine shock pulse - MIL-STD-810B Fig. 516. 1-2, Procedure III, Flight Vehicle Equipment
  A - Peak Acceleration - 30g
  B - Pulse Duration - 11 millisec.

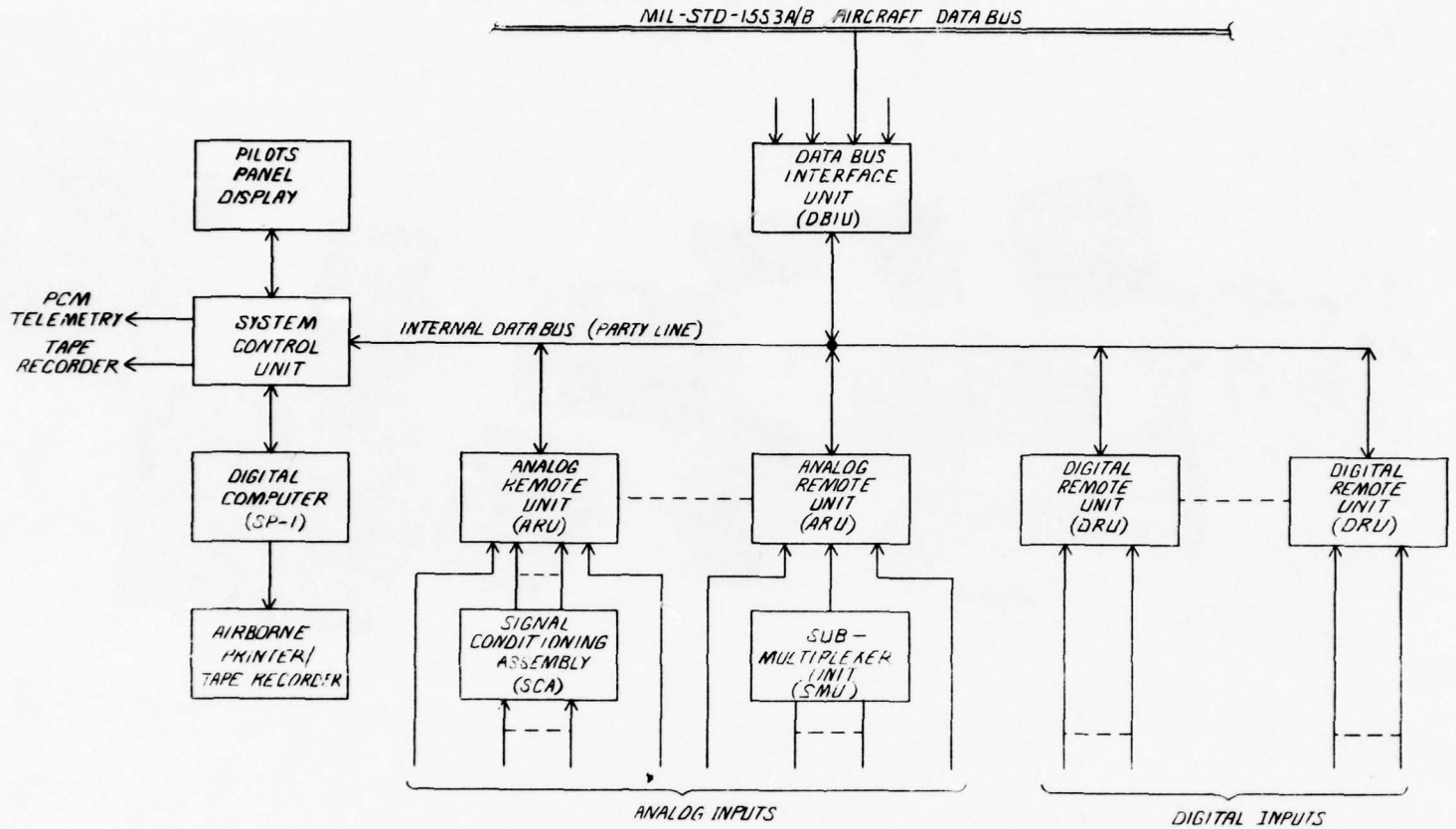
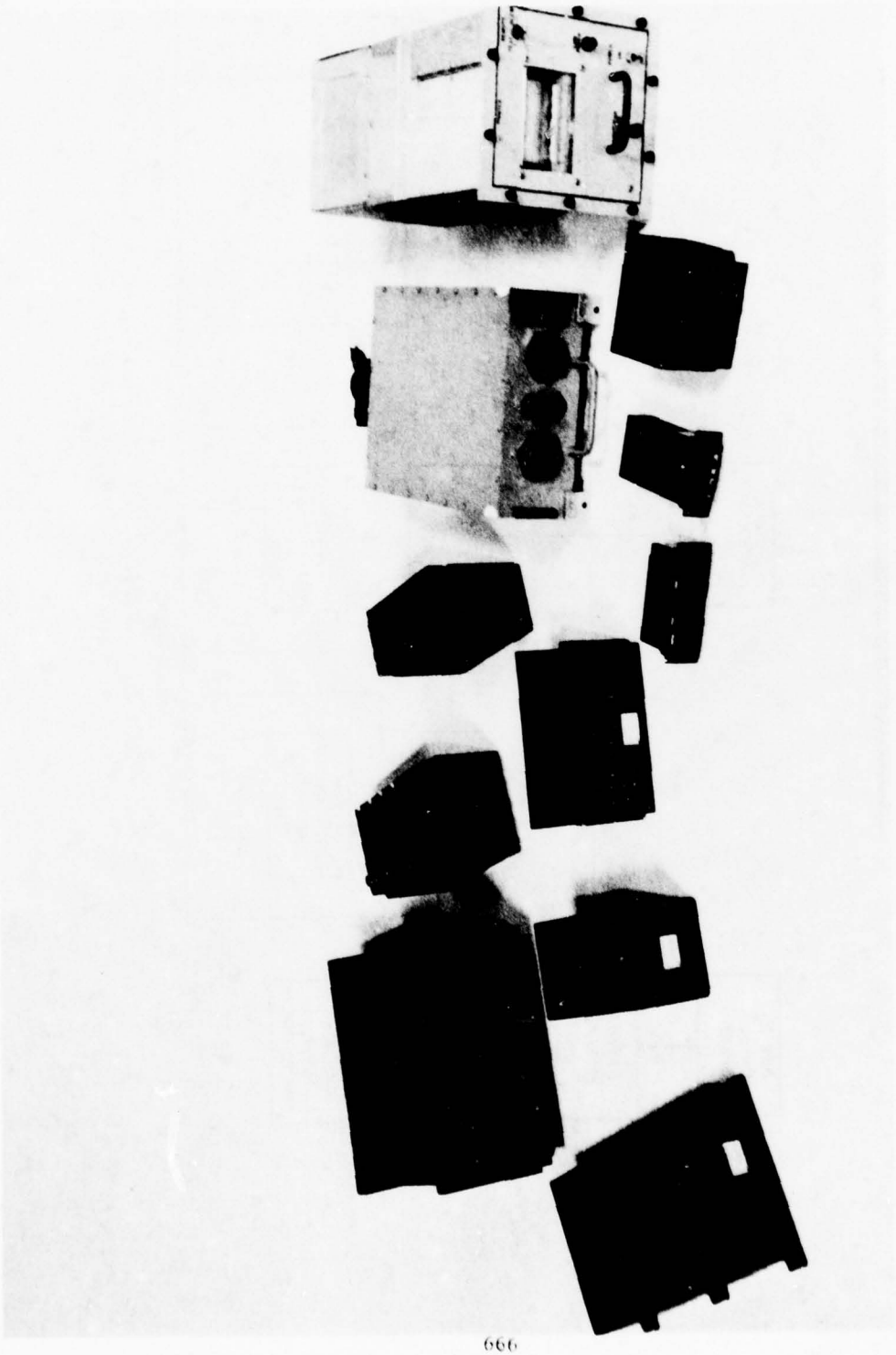


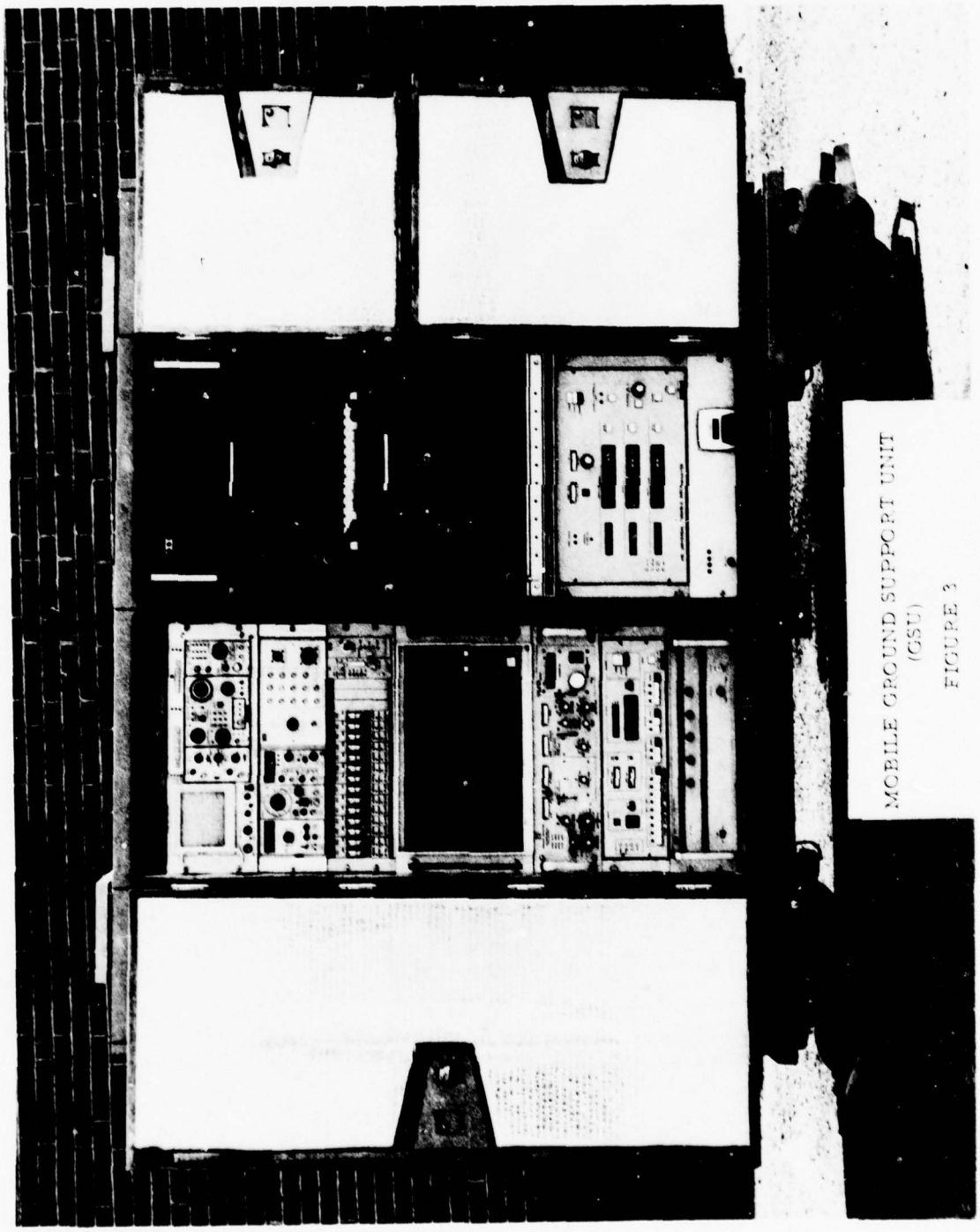

FIGURE 1 AIRBORNE TEST INSTRUMENTATION SYSTEM (ATIS) WITH MIL-STD-1553 BUS INTERFACE



AIRBORNE TEST INSTRUMENTATION SYSTEM
(ATIS)

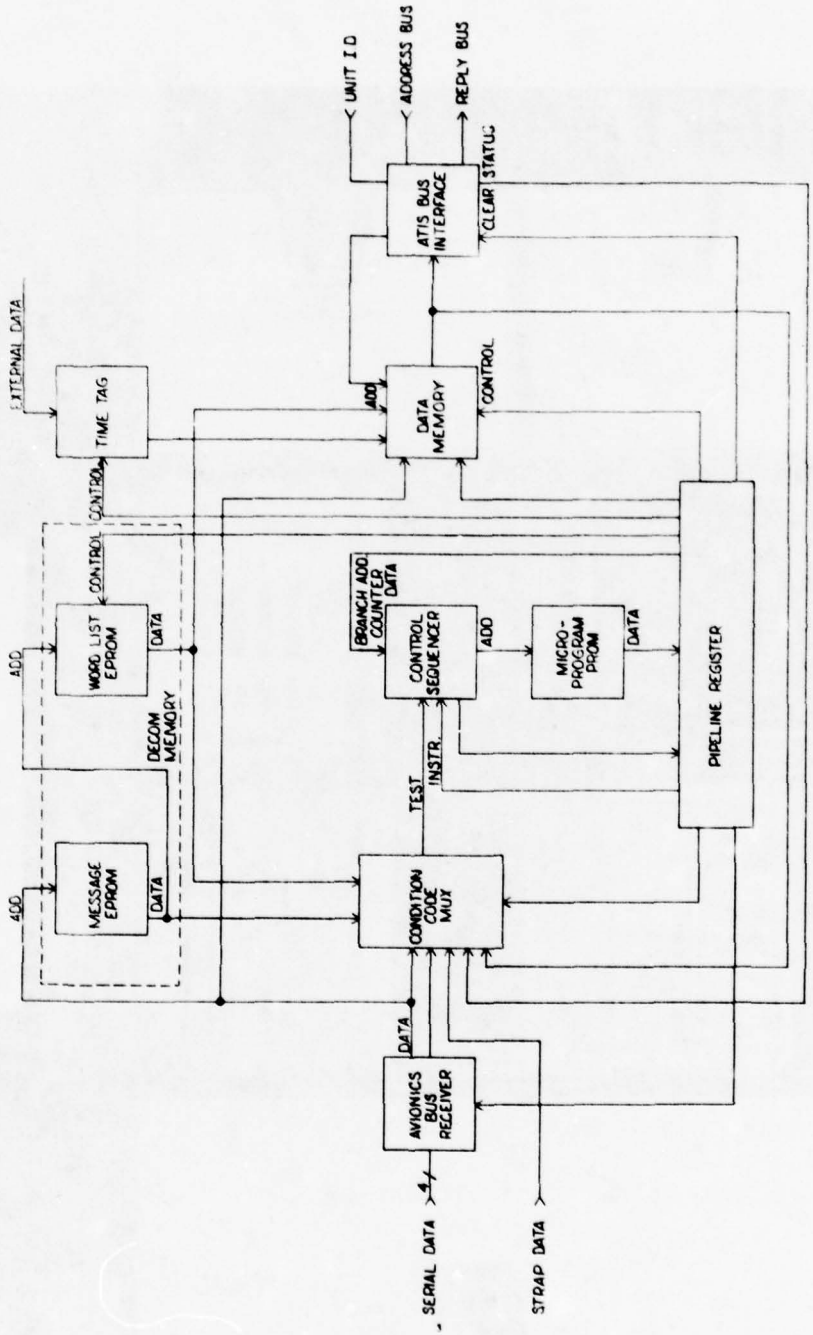
FIGURE 2

PRECEDING PAGE BLANK

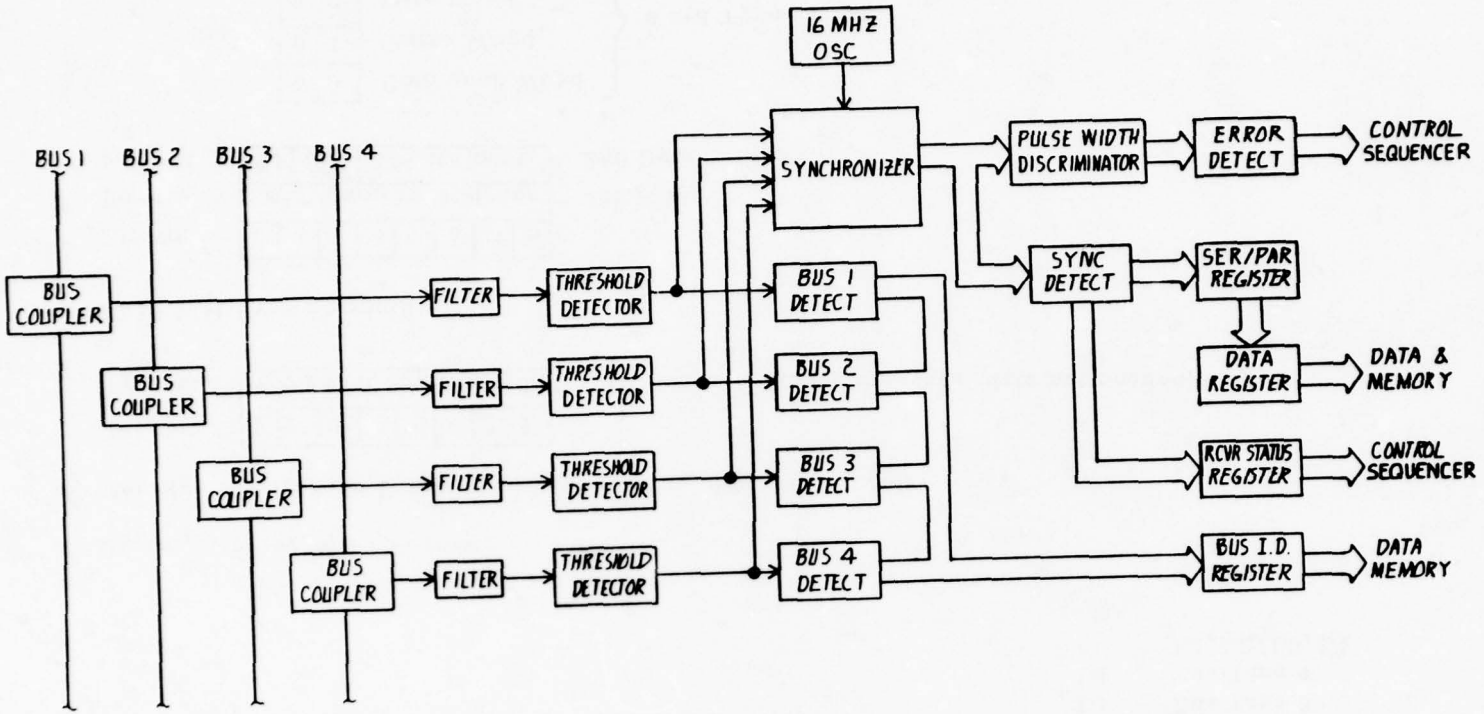


MOBILE GROUND SUPPORT UNIT
(GSU)

FIGURE 3

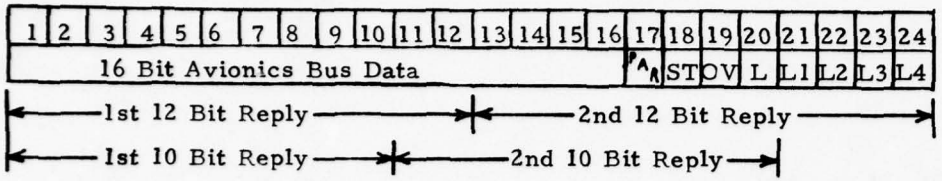



BLOCK DIAGRAM
DBIU
FIGURE 4




BLOCK DIAGRAM
AVIONICS BUS RECEIVER
Figure 5

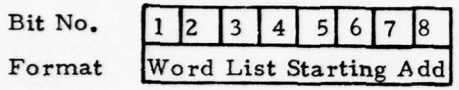
Data Memory Word Format




| | |
|---|---|
| PAR | Parity |
| ST | Staleness |
| OV | Overflow |
| L1 | Bus Line 1 |
| L2 | Bus Line 2 |
| L3 | Bus Line 3 |
| L4 | Bus Line 4 |
| L | L1, L2/L3, L4 |

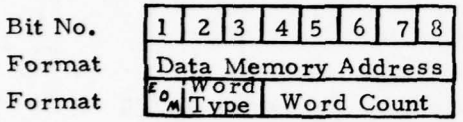
Decom Memory Word Formats

Message Recognition Byte (Word List Starting Address) Format




All "ones": Ignore message (data not needed)

Word List Byte Formats




| | |
|---|---|
| 0 0 | Command Word |
| 0 1 | Status Word |
| 1 0 | Data Word |
| 1 1 | Time Tag |

Word Type Codes

DBIU WORD FORMATS
Figure 6

# LOGIC STRUCTURE FOR TESTABILITY AND FAILURE DETECTION

U. Schulz and A. Roelker

DORNIER SYSTEM GMBH, GERMANY

ABSTRACT

A concept of the logic structure for testability and failure detection of digital, modular, onboard data acquisition and control units within avionics systems will be presented. The units of the avionics system are interconnected by the Serial Data Bus (1553).

The concept was developed according to the maintenance levels within the German Air Force. The tests and failure detection are performed during operation as well as during pre- and post flight tests without additional external special to type test equipments. The tests are totaly integrated into the avionic system.

The tests result in the localization of the failure on the defective module within a unit. The test-software needs about 4 msec and may therefore incorporated into the operational software.

This leads during operation(inflight)to a reorganization of the system by software such that the full system capability will be kept until a second failure occurs.

This system capacity is performed by a minimum of additional hardware (5 %) and testsoftware (10 %).

## 1. INTRODUCTION

The conventional structure of onboard electronic and avionic system is purely a collection of units and subsystems (for example: navigation, guidance and control, communication, displays, ...) which are separated and independent from each other. The failure detection and testing of such systems is performed by many special to type testequipments for pre- and postflight tests as well as for maintenance tests. Testing during operation is nearly impossible except the monitoring of a failure of a complete subsystem.
A concept for the testability and failure detection of integrated, digital onboard electronics within avionic systems is presented. The system makes use of the inherent redundancy.

## 2. ASSUMPTIONS

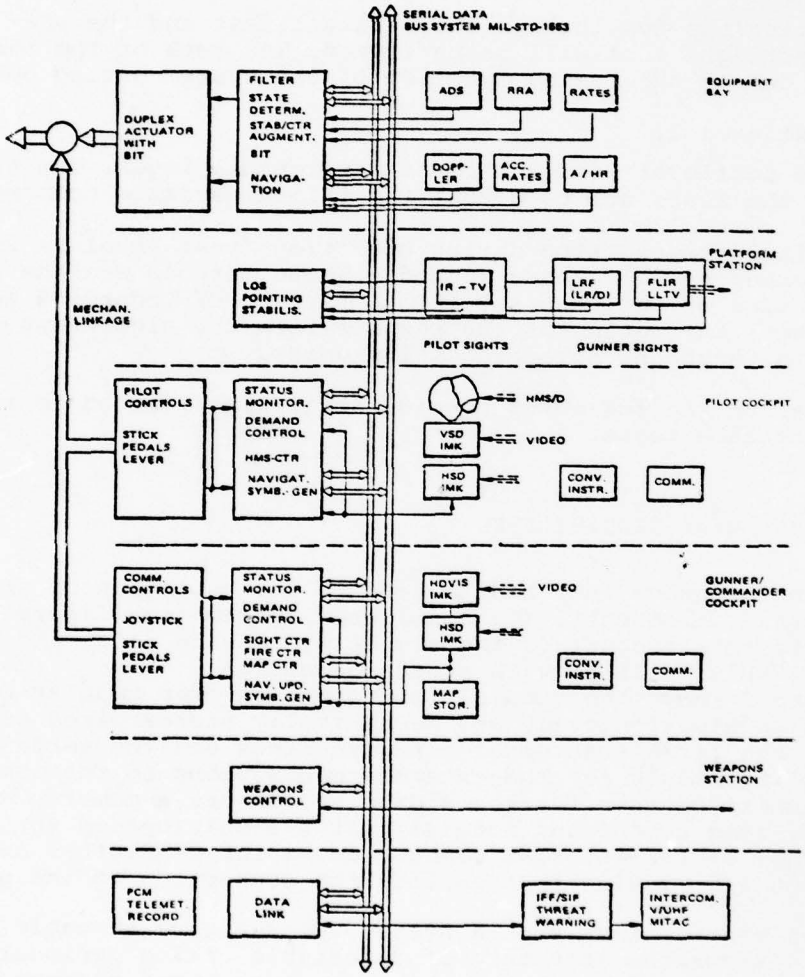
The following assumptions are made:

- The avionic system is structured as an integrated data processing system with a uniform structure concerning the signal processing soft- and hardware.
- The system contains partially redundant units due to the specified reliability of the subsystems.
- The signal processing units are line replaceable units (LRU) and interconnected by the serial data bus according to MIL-STD 1553.
- The internal structure of the LRU's is modular with the modules connected and operated via a parallel data bus. The modules are functional electronic boards as multiplexer, A/D-converters, digital inputs and outputs, ...
- Most of the LRU's are controlled by an internal processor.

The following approach on testability and failure detection is based on the system shown in figure 1. This is an example of an avionic system design for a helicopter for which the previous mentioned assumptions are valid.

Data Bus Interface Unit - Data Bus Interface Unit (DBIU) is the [illegible] chosen for the ATIS MIL-STD-1553 monitor/decom interface unit. The






DORNIER FIG. 1: **AVIONIC SYSTEM DESIGN ALTERNATIVE: MECHANICAL BACK UP**

## 3. CONSIDERATION OF THE PROBLEM

The failure detection and testing is divided into two main levels, according to the so called maintenance levels of the German Air Force, figure 2:

o Testlevel 1:

  During the testlevel 1 the inflight test and the pre- and postflight test will be performed. The task of the tests are to detect the failed function of the system during operation.

o Testlevel 2:

  The testlevel 2 is the first maintenance level. The tasks of the tests are to detect the failed hardware components.

The failure detection during operation (test level 1) is performed by well known methods. Those methods are the observation of the process, the comparison of redundant sensor-signals, the voting of the results form the algorithms within redundant data processing units.

Therefore the following considerations are limited to the maintenance tests.

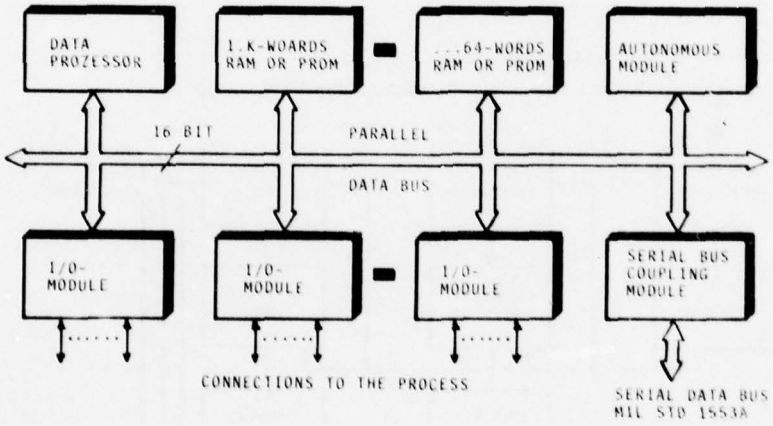
## 4. HARDWARE TESTABILITY

The maintenance test has to lead to the detection of the failed hardware components. The considered system (see figure 1) has a uniform structure of the signal processing soft- and hardware, with LRU's built up with standard modules.
Figure 3 shows the general configuration of a modular LRU with an electronically and mechanically standardized parallel data bus for the internal data transfers and the serial bus coupling module for the external connections to the other units. The units contain further a data processor, a memory, RAM or PROM, some autonomous modules that are only needed for the control of the internal events, and a lot of digital and analog I/O-modules that perform the connection to the process.

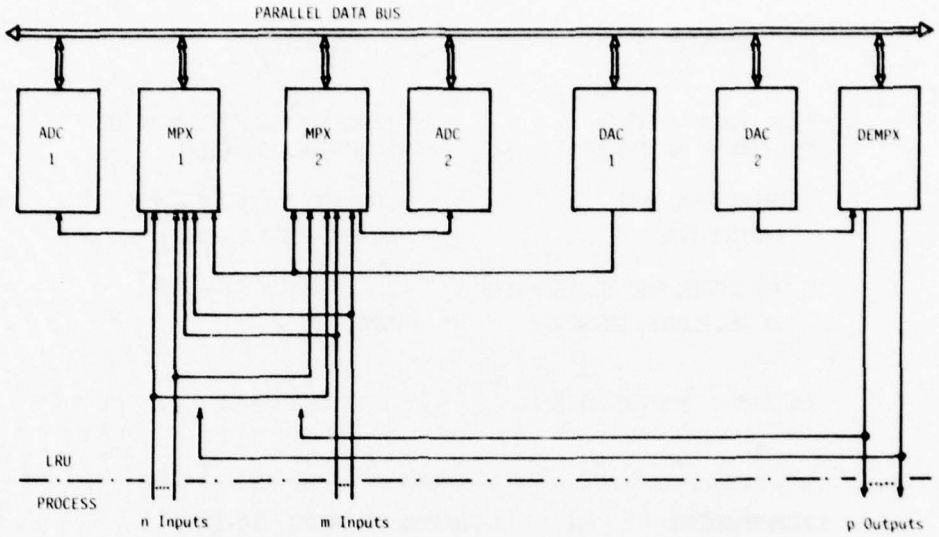
If we want to test such a system we must have testable hardware. Fig 4 shows the structure of a testable analog peripherie. We get the testability by using the inherent redundancy of the modules. The n- and m-input signals are given into the both

| TESTLEVEL 1 | TESTLEVEL 2 |
|---|---|
| o DETECTION OF A FAILED FUNCTION OF THE SYSTEM | o DETECTION AND LOCALIZATION OF THE FAILED HARDWARE COMPONENTS |
| OPERATIONAL MODE | SUSPENDED OPERATIONAL MODE |
| LIMITED TIME | NO TIME RESTRICTIONS |
| THE SYSTEM MUST BE CONTROLLED BY THE PROCESS SOFTWARE | THE SYSTEM IS IN A STABLE CONDITION |
| INFLIGHT PRE/POST-FLIGHT | MAINTENANCE |

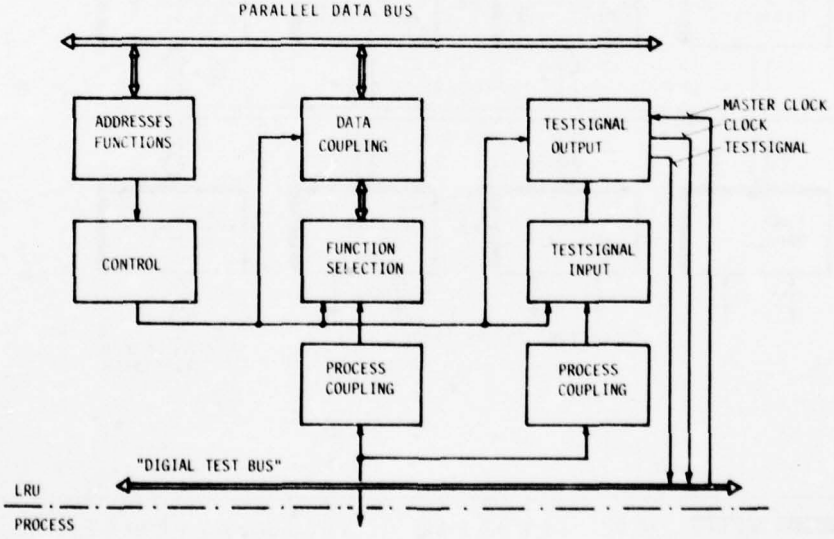
DORNIER FIG 2: TESTLEVELS FOR AVIONIC SYSTEMS




DORNIER FIG.3: GENERAL CONFIGURATION OF A LRU MUDAS-SUBSYSTEM




**DORNIER** FIG. 4: STRUCTURE OF A TESTABLE ANALOG PERIPHERY




**DORNIER** FIG. 5: STRUCTURE OF A TESTABLE DIGITAL I/O-MODULE

input modules and are given back again to the system by additional wiring from the outputs to the inputs (cross strapping). The test itself than is performed by software such that a failure within the analog peripherie may be detected and localized down to the level of analog input or output modules.

The figure 5 shows the structure of a testable digital I/O-module. Here some circuits are added to the pure module function of every module in order to set up the test information and to send it on a "digital test bus". The test information is converted to a serial 4-bit signal.

The figure 6 shows the receiver module that is connected to the parallel memory bus of the data processor. The test information than is transfered directly to the registers of the data processor and there compared against the measurement value that comes directly over the data bus from the module.

The additional hardware consists only in one receiver module, the circuits on the electronic boards and the wiring of the serial test bus. The tests itselves are made by software.

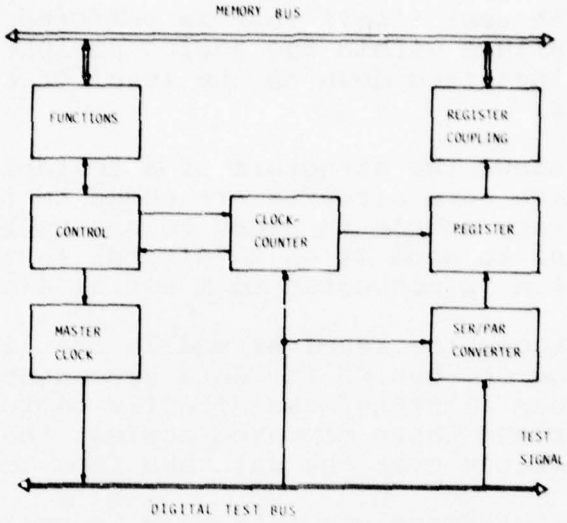
Figure 7 shows an example of a realized LRU within an experimental flight guidance and control system. The LRU is prepared for the maintenance test.

The analog periphery is tested as described before. The additional wiring (cross strapping) was realized as shown in figure 4. The test of the analog periphery includes the filtering modules and phase selective rectifiers.

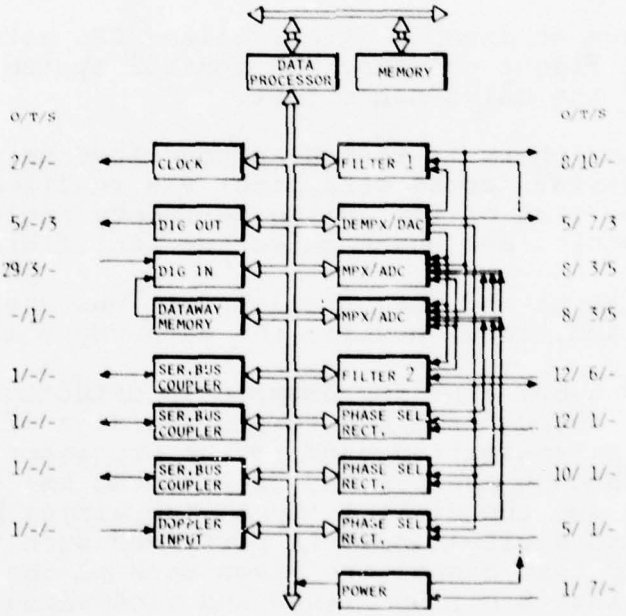
The digital input and the parallel data bus are tested by means of an additional module, the data way memory.

A failure of a bus line for example is detected by transmitting a testsignal through the bus to the register of the data way memory. It is transmitted back to the processor at one way through the bus and at a second way via the process-side and the digital input. The wiring between data way module and digital input is performed such that the first 8 bits of the test signal are given back to the processor through the last 8 bit bus lines and vice versa.

The test of the LRU consists further of short tests of autonomous modules (clocks, ...), the data transfer via the serial bus, the processor itself, the memory and power modules inside the LRU.




DORNIER FIG. 6: BLOCKDIAGRAM OF A TESTSIGNAL RECEIVER MODULE




INPUT AND OUTPUT SIGNALS: OPERATIONAL/TEST/SPARES (O/T/S)

DORNIER FIG. 7: LRU OF A FLIGHT GUIDANCE AND CONTROL SYSTEM WITH TESTABLE PERIPHERY

From investigation and tests we got the following results.

Considering the results of all tests and their logical combination the test leads to a localization of the defective module.

The calculation of such a test takes about 500 command words and a time of about 4 msec.

This leads to the possibility to perform the test at every calculation cycle of the processor while the system is full operating.

Due to this capability of failure localization at the level of the module during the operational mode it is possible to reorganize the software of a LRU after the detection of a failed module. The information for example coming from an analog input may be replaced in case of a failure of the module by a value that is calculated from redundant informations within the system (additional observer).

Figure 8 shows the block diagram of the software structure of a LRU. The test software now is incorporated into the operational software. The chosen test procedure depends on the system configuration. The organisation and timing of the complete software is performed by a task sceduler.

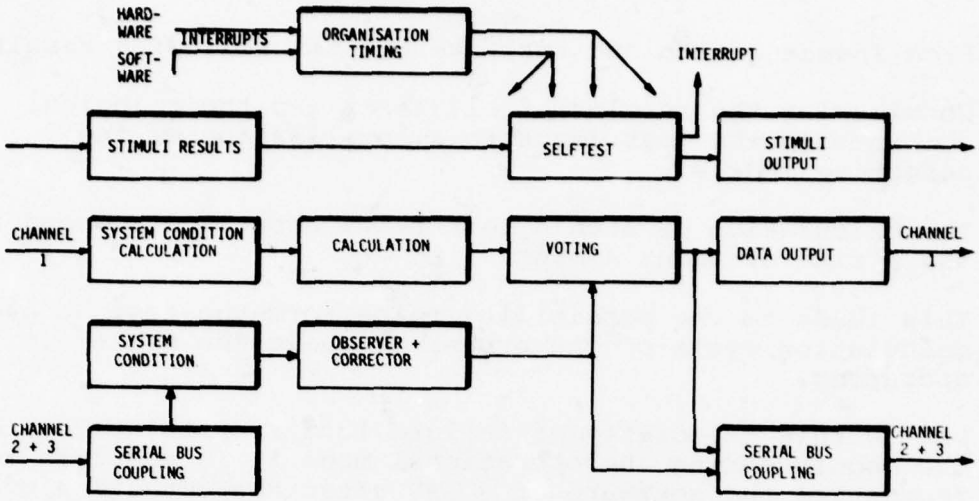
After the discussion of the testability of one LRU the maintenance test of the whole system (see figure 1) is considered.

In principle there are two possibilites to realize the maintenance test:

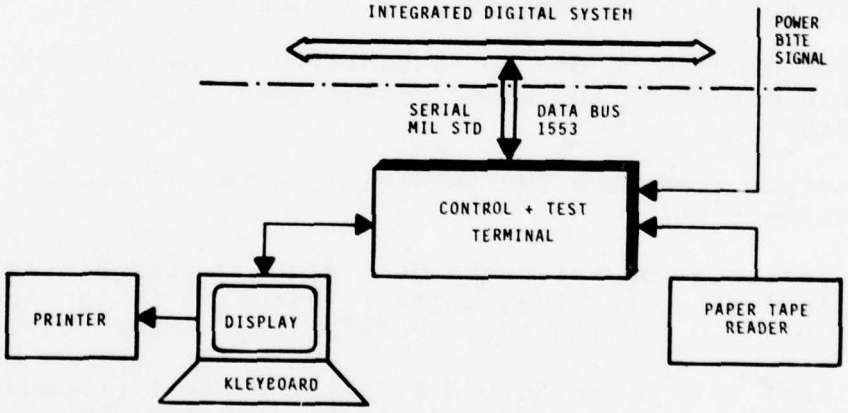
1. The test will be performed by a simple external test terminal with the test software situated in the memory of every LRU and the task sceduler in the terminal.

2. The test will be performed by internal test procedures that are situated completely within the memories of the LRU's and performed and controlled by one LRU being the master on the serial data bus.

An example for the configuration of an external test terminal is shown in the figure 9. The whole system is connected to the control and test terminal (CTT) via the standard interface of the serial data bus. With this




DORNIER FIG. 8: STRUCTURE OF THE SOFTWARE OF A LRU




DORNIER FIG. 9: EXTERNAL TEST LEVEL - 2 - TESTEQUIPMENT

configuration the whole system may be tested with the CTT being the master on the bus. On principle the same configuration is valid if the test will be performed by internal test procedures. In this case the communication between the system under test and the operator may be performed via a multifunction display and a multifunction keyboard in the cockpit (see figure 1).

## 5. FINAL CONSIDERATIONS

Inflight-, pre- and postflight-test as well as the maintenance test according to the testlevels 1 and 2 may be performed and totaly integrated into the system.

There is the possibility to detect the failed function of the system and to localize the failure at the level of the module within the LRU.

There is no need of any external test equipment if the result of the failure detection and the test may be indicated (immediately or on request) on a multifunction display in the cockpit. The decision whether the information of the failure must be indicated immediately or on request depends upon the importance of the failure and its influence at the operational mission during the flight. During pre-, postflight and maintenance test all results might be indicated.

With respect to the testable hardware and the methods of failure detection within redundant systems it is possible to allow one detected failure without degradation during operation (inflight). The software of the failed LRU will be reorganized as discribed above. The full system capability will be kept until a second failure occurs.

This is especially interesting for example in the case of a triplex guidance and control system.

This capacity of the system is performed by a minimum of hardware (5 %) and the test software (10 %) incorporated into the operational software program of the system.

# DIGITAL DATA ACQUISITION SYSTEM (D-DAS) — A FLEXIBLE TOOL FOR SIMULATION, DEBUG, and TRAFFIC MONITORING OF 1553 SYSTEMS

By: Michael Boice

EMM/SESCO
Severe Environment Systems Company
A subsidiary of
Electronic Memories & Magnetics Corp.

The Severe Environment Systems Company (SESCO) has developed a highly sophisticated and extremely flexible instrumentation system for numerous 1553 applications. As a passive monitoring system, D-DAS is capable of on-line data compression, thereby eliminating the recording of insignificant data and reducing the time necessary for interpretation of test results. This system finds ready application in the testing and debug of 1553 systems and is also usable as a mass memory peripheral.

PRECEDING PAGE BLANK

The proliferation of the MIL-STD-1553 Bus has greatly simplified the integration of avionics systems, and has provided a sufficient product base to warrant the development of specialized systems hitherto impractical due to prohibitive engineering costs. This is well exemplified by EMM/SESCO's D-DAS product, a highly sophisticated system peripheral capable of numerous functional activities while utilizing a common hardware configuration.

EMM/SESCO's D-DAS, as shown in Figure No. 1 - Block Diagram, consists of a microprogrammed 1553 interface, the internal SECS-2 16 bit minicomputer, a ROM 4K x 16 control store, interface logic, a tape controller and two 23 megabit SETS-1 tape systems. These components, including power supply, are housed in a 1/2 ATR enclosure, Figure No. 2, with individual modules conduction cooled by integral heat-sinks to the side walls. Power can be 28 vdc or 110 vac 50-400$H_Z$. The system conforms to MIL-E-5400 specifications.

The D-DAS 1553A interface contains the dual bus interface circuitry, the serial to parallel converters, a DMA interface to the internal EMM BUS, and eight 2K x 1 message selection tables. The electrical components include proven transformer and standard hybrid transmit/receive devices. The encoding and decoding of the Manchester signal is accomplished utilizing the HDK-15530-8, manufactured by Harris Semiconductor. This device also accommodates the checking of word count validity and provides a convenient means of serializing and deserializing the 16 bit command, data, and status words. An AMD 2911 sequencer is the central element of the 1553 command sequencer. A 1K x 16 control ROM is utilized to implement the synchronization algorithm, the required 1553 protocol and the response to mode commands.

The SECS-2 minicomputer card set is a microprogrammed minicomputer. Included is the capability to perform hardware fixed point arithmetic and to allow direct implementation of multiply, divide, and multiple shifting. Double precision 32-bit words can also be handled. The four basic single precision, floating point arithmetic operations are also programmed. The standard system offers a modified I/O and interrupt structure (EMMBUS) suitable for efficient data transfer. A compatible I/O is optionally available for hook-up to existing standard peripherals. The basic card set consists of four 6½'' x 4½'' plug-in cards. The four cards include: (1) the Microcontrol Card, (2) the Data Card, (3) Logic Card A, and (4) Logic Card B.

The SETS-1 is a high-performance digital data storage unit developed by SEPD for severe environment military, aerospace, and industrial applications. The basic SETS-1 consists of a compact transport module and a removable, sealed tape module. Housed in the transport module are drive and I/O circuits, and the solid-state D.C. brushless motor. The environmentally-sealed tape module, interchangeable between transports, contains 300 feet of wide temperature tape, a four-track Read/Write head, and coaxial reels with capstan drive, tape guides, and tape position sensing. Data can be recorded at a density of 1600 bpi for 23 Mbits of storage.

The primary utilization of the D-DAS hardware configuration is as a passive bus monitor for flight test and system debug applications. In this role, a complex microprogrammed algorithm[1] traces each 1553 message to maintain the D-DAS in proper synchronization, i.e., not confuse a status word with a command word.

1. Synchronization algorithm and recorded data format developed in conjunction with Captain L. S. Dougherty AFAL/AA/DAIS.

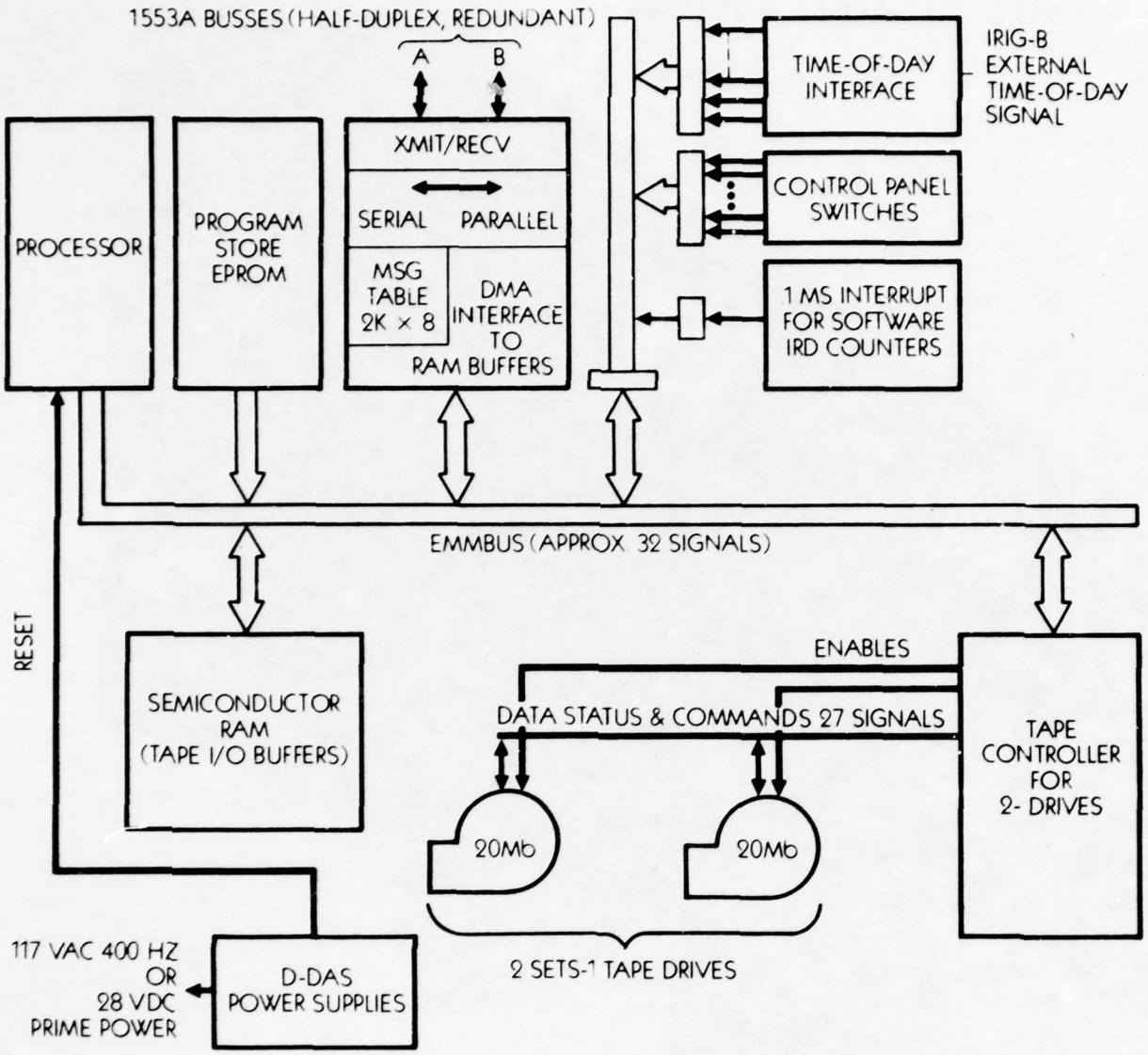



**FIGURE 1 D-DAS SYSTEM BLOCK DIAGRAM**

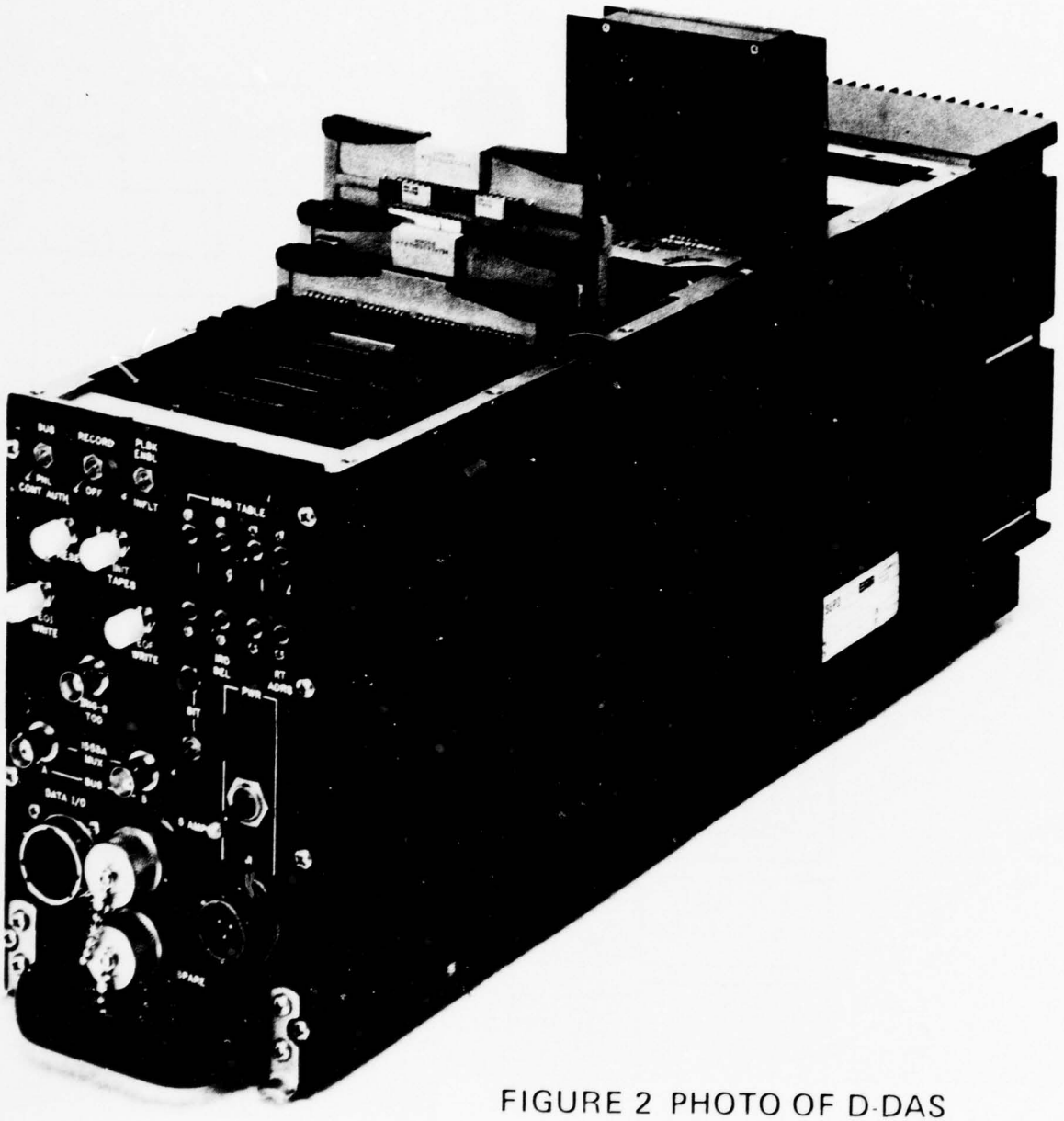



FIGURE 2 PHOTO OF D-DAS

When a command word is encountered which matches one which the user programmed as significant, the 1553 interface transfers the message, along with a sixteen or 32-bit time of day tag into its 4K x 16 RAM memory. This process is repeated until a data record approximately 2K x 16 in length is established. This record is then transferred to the SETS-1 magnetic tape recorders contained in D-DAS.

The D-DAS 1553 interface contains a single 2716 type, socketed EPROM, which is user programmed to establish the appropriate message selection tables. This enables the pre-programming of eight 2K x 1 message tables which are then selectable via front panel control or by the system bus controller.

When recording data in the specified format of Figure No. 3, the D-DAS shall be capable of recording 1300 records of information where each record contains 2048 by 16 bit words, with a maximum total storage of 46 million bits. The system is capable of transferring data to or from the SETS-1 at a rate of 48,000 or 192,000 bits per second as selected by front panel switch settings.

To provide a Range Time reference, D-DAS is capable of accommodating an external IRIG-B format time-of-year code generator. D-DAS decodes this serial PCM input and utilizes it to time tag the messages received with absolute range time. As an additional feature, the D-DAS also acts as the RT for the time-of-year generator which can therefore be accessed via the 1553 Bus as a D-DAS subaddress.

While in the passive operational mode, D-DAS is initialized by its front panel controls to its RT address, functional mode, and appropriate user message table. In this mode, there is absolutely no reconfiguration or modification necessary to the host system and the D-DAS is totally transparent to the systems being monitored as well as the bus controller. The user thus need make no hardware or software modifications to the system under test other than the physical connection of the D-DAS to the 1553 bus.

While this mode of operation represents a significant advancement in system instrumentation, the user can achieve substantially more efficient D-DAS utilization by incorporating minimal changes in bus controller software. When the Control Authority switch on the D-DAS front panel is in the bus position, the D-DAS control word can be accessed and modified via the 1553 bus. This enables the bus controller to start and stop the monitor function, switch the active message tables and even preblock and send data to the D-DAS. Thus, the user may activate the D-DAS only during significant portions of a test, and select the appropriate message selection table for each test phase.

As the aforementioned 2K x 16 records are accumulated, they are formatted into records as shown in Figure No. 3. The initial message is preceded by a 32-bit time of year tag as obtained from the external time code generator and all subsequent messages are preceded by a 16-bit interval time in relation to the first message. This format, developed in conjunction with W-PAFB[1] allows for easy interpretation of data and also enables one to integrate composite records in applications where multiple D-DAS units are used.

1. Synchronization algorithm and recorded data format developed in conjunction with Captain L. S. Dougherty AFAL/AA/DAIS.

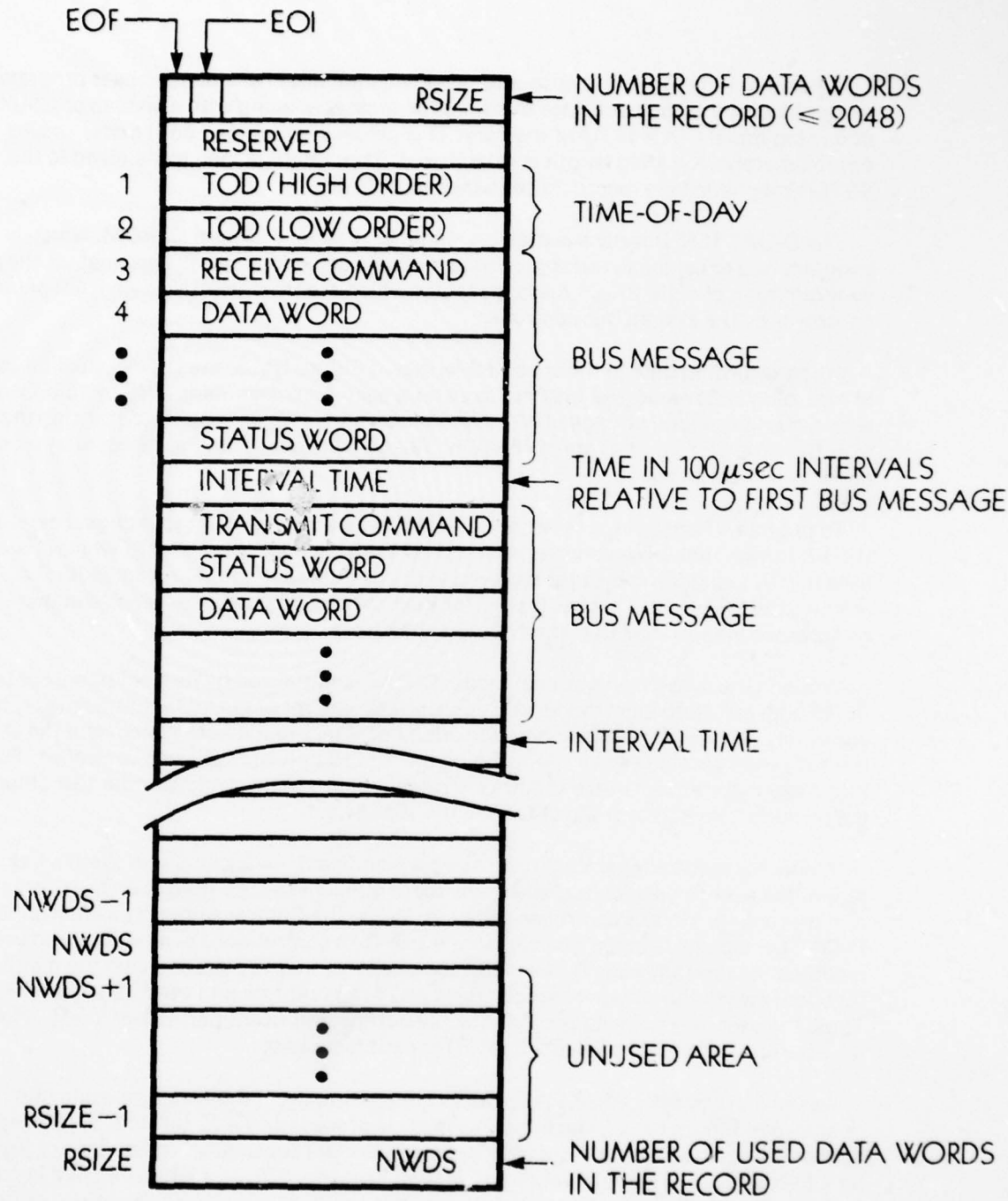
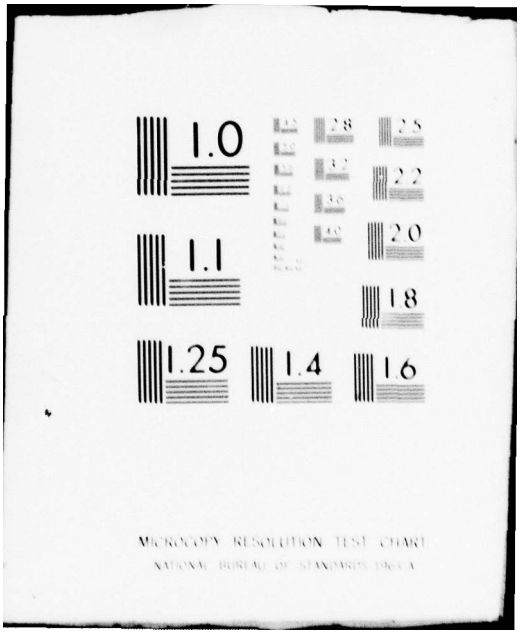



**FIGURE 3 RECORD DATA FORMAT**

AD-A068 310 AERONAUTICAL SYSTEMS DIV WRIGHT-PATTERSON AFB OH F/G 17/2
PROCEEDINGS OF THE AFSC MULTIPLEX DATA BUS CONFERENCE (2ND), HE--E
OCT 78 E C GANGL
UNCLASSIFIED ASD-TR-78-34-VOL-2 NL

8 OF 9
AD A068310




1.0
1.1
1.25
1.4
1.6
2.8
2.5
2.2
2.0
1.8
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

Ground based data transfer can be achieved by one of two methods. Either the entire D-DAS unit can be removed from the test vehicle and the data removed via the 1553 bus utilizing the protocol defined by the B52 OAS/DTU, or only the SETS tape cartridges removed and a ground based SETS recorder used to playback the data (interfaces are available for the SETS-1 to several commercial minicomputers). EMM/SESCO also envisions development of a "quick look" ground station utilizing a minicomputer, a nine track 1/2 inch tape unit, a video, and a floppy disk which will not only be capable of transferring the data to a nine track format for analysis but will also be user configurable such that test data can be immediately examined, by parameter, in engineering units easily interpretable by test personnel.

While the advantages of D-DAS in instrumentations and debug of 1553 systems is patently obvious, the development of a "quick look" ground station can extend its function beyond the flight test itself to where D-DAS can be utilized for both training and pilot evaluation. The evolution of "realistic" training missions has greatly complicated the evaluation of pilot performance to the point that extensive monitoring systems are necessary. Without rapid and accurate feedback, training missions are very limited in being able to improve individual pilot performance. A partial solution to this problem has been implemented in video recording of the heads up display in modern training. This, however, cannot achieve a total level of cognizance with regard to pilot interaction with, and function of, the avionics system. Utilization of a D-DAS tape system with a "quick look" ground station capable of integrating both 1553 and video data would provide a level of sophistication which will dramatically improve the training feedback available and thus accomplish the training function in a more expeditious and cost effective manner.

Another application of the D-DAS hardware set which is a natural outgrowth of the passive bus monitor function is that of a 1553 mass memory peripheral. The protocol established for the playback of prerecorded tapes via the multiplex data bus also provides a set of tape function commands such that the unit can be accessed as a Mass Tape Memory by either the bus controller or other RT. The D-DAS can therefore be used for both Initial Program Load, Program and Data storage for any or all systems on the data bus. The removable SETS-1 cartridges provide a convenient method by which programs and data can be transported between ground loaders and the on board avionics system.

The flexible D-DAS hardware set is capable of satisfying numerous functions and is applicable to any system utilizing the 1553 multiplex data bus. This paper has described several of these including the passive data bus instrumentation system, and a mass memory peripheral. As the internal D-DAS processor may be user programmable, its ultimate applications are limited only to the imagination of the system integrator or end user.

# DATA BUS SIMULATION AND TESTING ON THE SPACE SHUTTLE PROGRAM

Michael R. Sottile
Singer-Kearfott Division
Little Falls, N.J.

## ABSTRACT

This paper discusses the analysis and performance verification of the Space Shuttle Data Bus configurations. The analytic approach for the development of a Word Error Rate model is discussed and typical results are presented.

To verify the analytic results and to substantiate mux system performance, several Space Shuttle Data Buses were constructed using MIA's and the Space Shuttle Data Bus Cable and tests performed; among many others, these included WER and waveform fidelity. Also, data was taken to determine how Word Error Rate varied parametrically with signal rise time, signal offset, internal threshold voltages, rms noise levels, signal offset, temperature and signal amplitude.

PRECEDING PAGE BLANK

## INTRODUCTION

The Space Shuttle Multiplex System represents a milestone in the Avionics industry. It is the first large scale application of party line data multiplexing. The critical nature of this communication makes necessary the minimization of error. To verify that the data error rate was within an acceptable limit both data bus system analysis and verification tests were performed.

## SYSTEM REQUIREMENTS

All communications are via a fixed word structure - a 3 μsec sync field followed by 24 data bits and a single parity bit. The data code is Manchester II (Bi $\phi$-L) per MIL-STD-442 at a bit rate of 1 megabit per second. The sync field is a nonvalid Manchester code. The positive and negative pulses that form the sync are 1.5 μsec wide. A command word sync is formed by a positive pulse (1.5 μsec wide) followed by a negative pulse (1.5 μsec wide) and a data word sync is formed by a negative pulse followed by a positive pulse.

Figure 1 illustrates a typical Data Bus configuration. Each host LRU contains a Multiplexer Interface Adapter (MIA) for data communication over the data bus. The MIA provides data validation (error checking) logic to assure that the received word conforms to the following minimum validation criteria:

a. First a valid sync field must be received. If the sync field is invalid the MIA does not respond to the data field.

b. Each bit in the data field is a valid Manchester II code. If any bit in the data field is declared invalid a nonvalid Manchester error flag is raised (for use by the Host LRU).

c. The word has a minimum of 24 data bits plus parity. If fewer than 25 bits are received a bit count error flag is raised.

d. The parity bit renders the complete 25 bit data field "odd". If a parity error is detected a parity flag is raised.

When a word received by a MIA fails to conform to these criteria the word is considered invalid and a detected word error (also called a bit miss) is declared.

Figure 2 illustrates the operation of the SMIA and especially shows the error signals available at the host interface. When the host receives the Rcvr Bus Active signal a valid sync has been detected. At the End of Word strobe the user interrogates the error signals NV Manchester, Bit Count and Parity Error to verify error free detection of the incoming word.

The Space Shuttle Orbiter requires that at any stub the Word Error Rate (WER) be $3X10^{-7}$ or better (less) when:

a. The signal amplitude is 2 volts peak.

b. The noise is 300 mv RMS white Gaussian distributed over the band of 1000 hz to 4 Mhz.

c. The signal rise time is 250 μsec.

The WER was defined to be:

$$WER = \frac{\text{Sync Misses + Bit Misses + Undetected Bit Errors}}{\text{Number of Words Transmitted}}$$

For the purpose of WER performance verification a Sync Miss is defined as the inability to declare a valid sync (start reception) when a valid sync is known to have occurred and an Undetected Bit Error is defined as an observed error in any bit position provided that a valid sync has been detected and a Bit Miss has not occurred.

Verification of this performance was accomplished by both analysis and test.

VERIFICATION BY ANALYSIS

To accomplish this by analysis it required that we develop:

a. Lumped parameter models of the SMIA in both the transmit and receive modes (see Figures 3 & 4)

b. A lumped parameter model of the Data Bus Coupler (Figure 5).

c. A model of the SMIA input filter (Figure 6).

d. A lossy transmission line model representative of the cable being used. By measurement of various lengths of the actual Space Shuttle cable the model illustrated by Figure 9 was derived.

Use of these models enabled the analysis of the total data bus system. The Space Shuttle mux bus configurations were reviewed and representative configurations were analyzed using Rockwell supplied input signals. This analysis determined waveform fidelity and WER performance for numerous stub locations on several bus configurations.

The WER Simulation Table (Table I) summarizes the range of variance, used for the WER analysis, for the input signal amplitude, the input noise, the signal rise time, the SMIA threshold voltage, & the SMIA filter bandwidth.

To get a measure of the affects of measurement error the input signal amplitude was allowed to vary by $\pm 3.5\%$ and the input noise (RMS value) was allowed to vary by $\pm 5\%$.

The WER was computed for all of the above conditions and the results are shown in Figures 10-20.

Waveform fidelity was determined by simulation of the various bus configurations. The results were later verified by tests on the identical bus configurations. Figures 21, 22 and 23 are representative of the results obtained by this simulation. The insert in these figures is the waveform obtained during test of the identical bus configuration.

VERIFICATION BY TEST

To verify the computer simulation results special test equipment was developed to simulate both normal and abnormal data bus operation. Tests were run to determine:

a. Effects of noise on bus operation.

b. Effects of data bus lengths, stub lengths, stub locations, line discontinuities, etc. and to verify performance such as:

c. Distortion tolerance.

d. WER

e. Design margin.

f. Negative test limits

g. Environmental effects.

WER SIMULATION - TABLE I

| PARAMETERS | RANGE OF VARIANCES | | | |
|---|---|---|---|---|
| SIGNAL S: | 1.5 | 1.75, | 2.0, | >2.0 VOLTS |
| NOISE N: | | VARIOUS | | MV |
| RISE TIME $t_R$: | 20.8, | 124.8, | 249.6 | NS |
| THRESHOLD VOLTAGE $V_T$: | 170, | 190, | 210 | MV |
| FILTER BANDWIDTH BW: | MIN | NOM | MAX | |
| SIGNAL OFFSET: | -50 TO 100 MV | | | |
| WORD PATTERN: | ALL 1'S, ALTERNATING 1'S AND 0'S | | | |
| SYNC: | COMMAND, DATA | | | |

To perform these tests the test configuration illustrated by Figure 24 was assembled. Since the noise generator provides very wide band (20 Mhz) white noise it was necessary to shape the noise spectrum to within specification limits (1 khz to 4 Mhz) prior to mixing with the signal. This was accomplished with mixer shown in Figure 25. This unit has the capability to process the input signal either shaped or unshaped prior to mixing with the band limited noise. When the shaped signal is desired the signal rise time amplitude symmetry and peak-peak amplitude are individually controllable.

Table II summarizes the capability of this Data Bus System Test System.

To establish confidence in both the computer simulations and in the Data Bus System test results whenever possible results were cross checked. Figure 26 shows the close correlation achieved between the computer simulation and the test results for WER as a function of *S/N*.

Table III summarizes some of the tests conducted during the Data Bus System evaluation program. These tests were run on several different bus configurations.

Typical of the Shuttle bus configurations is the LB1 bus (Figure 26). This is a bus of moderate length (146 ft), has 8 stubs and a 4 stub cluster at one end. Tables IV to VII summarize the results of the amplitude, sync field symmetry and SMIA clock frequency limit tests and the negative tests performed with this bus configuration.

## SUMMARY

This program has revealed that there are many factors to consider during the design of a data bus system. Aside from the usual considerations for voltage variations and temperature variation one must now be concerned about such parameters as waveform asymmetry, signal rise time at the receiving unit, wave shape distortion and wave shape offset.

All of these adversely impact the system WER performance. For a 2.5V pk signal (at the receiving MIA) the parameter variation impact on WER is:

- Bit Asymmetry (@5%) - 1 Order of Magnitude
- Threshold Voltage - ½ Order of Magnitude
- Filter Bandwidth - ½ Order of Magnitude
- Power Supply Variation - Affects threshold voltage & filter bandwidth
- Input Rise Time - A factor of 2 around nominal of 150 μsec

- Wave Shape Distortion - 2-3 Orders of Magnitude for extreme distortion
- DC Offset - At 150 mv noise, 2.6 V signal, a 25 mv DC offset results in a ½ Order of Magnitude degradation
- Temperature - 1 Order of Magnitude from -30°C to +92°C

The above WER impacts are for the Space Shuttle Data Bus System configured with Singer MIA's - other multiplexer designs would have to be separately analyzed to determine their parameter variation impact on WER.

From the analysis and data generated during this program it has been determined that the Singer MIA's satisfy the Space Shuttle Data Bus performance requirements.

DATA BUS SYSTEM TEST SET CAPABILITY

TABLE II

MODES OF OPERATION

- o ECHO
- o NORMAL

WORD RATE CAPABILITY

- o MAX RATE (3 μSEC WORD GAP)
- o 1 KHz RATE
- o OTHER VIA EXTERNAL STIMULUS

DATA WORD

- o FIXED (MANUAL INPUT)
- o VARIABLE (PSEUDO RANDOM)

LIMIT TESTING (DATA BUS SIGNAL)

- o SYNC FIELD
- o DATA BIT
- o DATA RATE (VIA EXTERNAL STIMULUS)
- o OUTPUT SIGNAL LEVEL

NEGATIVE TESTING

- o DROPPED BIT
- o EXTRA BIT
- o PARITY
- o INTERRUPTED TRANSMISSION

ERROR TESTING

- o WORD COUNT
- o MISS COUNT
- o BIT COUNT
- o DATA SAMPLING VALIDATION
- o 3 MIA SIMULTANEOUS TEST
- o NOISE GENERATOR EXTERNAL TO DBS T/S

HOST LRU (ADJUNCT TO DBS T/S)

- o MIA
- o LOGIC FOR ECHO MODE OPERATION (DB I/O)
- o STORAGE AND BUFFERS FOR NRZ OUTPUT TO T/S (DB IN/NAZ OUT)
- o BUFFERED CONTROL SIGNAL I/O INTERFACE

DATA BUS SYSTEM EVALUATION TESTS
TABLE III

| TEST | SIGNAL V-PK | NOISE mV rms | CLOCK MHz |
|---|---|---|---|
| WER VS P.S. | 1.5 | 300 | 16±.01% |
| WER VS RISE TIME | 1.5<br>2.0<br>2.5 | @300<br>@400<br>@550 | 16±.01% |
| DC OFFSET | AS-IS | 0 | 16±.01% |
| WER VS RECEIVER CLOCK | 2.0 | 300 | MIN, MAX |
| WER VS ASYMMETRY | 1.5 | 300 | 16±.01% |
| WER VS SIGNAL/ NOISE | 1.5<br>2.0<br>3.5 | VARY NOISE FOR ≥4 POINTS ON CURVE FOR EACH CONDITION. | 16±.01% |
| UNDETECTED BIT ERROR | AS-IS | 3 VALUES | 16±.01% |
| WER VS NOISE @ MDM TEMP | 2.0 | VAPY NOISE FOR ≥ 4 POINTS ON CURVE. | 16±.01% |
| LBI BUS WER TESTS | AS-IS | VARY NOISE FOR >4 POINTS ON CURVE AT EACH LOCATION. | 16±.01% |
| 20 HOUR WER TEST | AS-IS | 150 MV | 16±.01% |
| LBI BUS WER WITH DMIA | AS-IS | VARY NOISE FOR ≥4 POINTS ON CURVE. | 16±.01% |
| LBI BUS WAVEFORMS | AS-IS | -0- | 16±.01% |

DATA BUS SYSTEM EVALUATION TESTS
TABLE III (CONTINUED)

| SYNC SYMMETRY (% Δ) | POWER SUPPLY | TEST WORD | SYNC | TEMP | RISE TIME NS |
|---|---|---|---|---|---|
| NOM. | NOM + 4 | PR | DATA | LO, NOM , HIGH | 150 |
| NOM. | NOM + W.C. | PR | DATA | LO, NOM , HIGH | 100, 150, 200 |
| NOM. | NOM + W.C. | "1" "0" "10" "01" | DATA CMD | CYCLE | |
| NOM. | W.C. | PR | DATA | LO,NOM, HIGH | W.C. |
| +5, +3, -3, -5 | W.C. | PR | DATA CMD | LO, NOM , HIGH | NOM + W.C. |
| NOM. | W.C. | PR + OFFSET | DATA | LO, NOM, HIGH | NOM + W.C. |
| NOM. | W.C. | PR | DATA | LO, NOM, HIGH | AS-1S |
| NOM. | W.C. | PR | DATA | 91°C | NOM. |
| NOM. | W.C. | PR | DATA | LO, NOM, HIGH | AS-IS |
| NOM. | W.C. | PR | DATA | W.C. | AS-IS |
| NOM. | W.C. | PR | DATA | DMIA/SMIA HOT /HOT&COLD | AS-IS |
| NOM. | W.C. | "111" | CMD | ROOM | AS-IS |

NOM=NOMINAL
PR=PSEUDO RANDOM
W.C.=WORST CASE
CMD=COMMAND

TABLE IV. LIMIT TEST, AMPLITUDE

SMIA #1 = SER. #0002
SMIA #2 = SER. #0004
SMIA #3 = SER. #0008

| | | | | |
|---|---|---|---|---|
| Tx | 1 | | | |
| SMIA #1 | 7 | 6.4 | 3-88 * | MISS |
| | | | 3-89 ** | |
| SMIA #2 | 6 | 5.0 | 3-92 * | NO EOW |
| | | | 3-93 ** | |
| SMIA #3 | 8 | 6.4 | 3-90 * | MISS |
| | | | 3-91 ** | |
| Tx | 6 | | | |
| SMIA #1 | 7 | 6.4 | 3-96 * | MISS |
| | | | 3-97 ** | |
| SMIA #2 | 1 | 5.0 | 3-94 * | NO EOW |
| | | | 3-95 ** | |
| SMIA #3 | 8 | 6.4 | 3-98 * | MISS |
| | | | 3-99 ** | |
| Tx | 8 | | | |
| SMIA #1 | 3 | 6.4 | 3-102 * | MISS |
| | | | 3-103 ** | |
| SMIA #2 | 1 | 5.0 | 3-104 * | NO EOW |
| | | | 3-105 ** | |
| SMIA #3 | 6 | 6.4 | 3-100 * | MISS |
| | | | 3-101 ** | |

* 5us/cm .5V/cm
** 1us/cm

TABLE V. LIMIT TEST, SYNC FIELD SYMMETRY

SMIA #1 = S/N 0002
SMIA #2 = S/N 0004
SMIA #3 = S/N 0008

| | | SYNC FIELD SYMM. (μS) | | | | |
|---|---|---|---|---|---|---|
| | | COMMAND SYNC | | DATA SYNC | | ERROR |
| | STUB LOCATION | 1ST ½ FIELD | 2ND ½ FIELD | 1ST ½ FIELD | 2ND ½ FIELD | TYPE |
| Tx | 1 | 1.65 | 1.35 | 1.65 | 1.35 | NO EOW |
| SMIA #1 | 7 | 1.4 | 1.6 | 1.4 | 1.6 | MISS |
| SMIA #2 | 6 | 1.65 | 1.35 | 1.65 | 1.35 | NO EOW |
| | | 1.4 | 1.6 | 1.4 | 1.6 | MISS |
| SMIA #3 | 8 | 1.65 | 1.35 | 1.65 | 1.35 | NO EOW |
| | | 1.4 | 1.6 | 1.4 | 1.6 | MISS |
| Tx | 6 | | | | | |
| SMIA #1 | 7 | 1.65 | 1.35 | 1.65 | 1.35 | NO EOW |
| | | 1.4 | 1.6 | 1.4 | 1.6 | MISS |
| SMIA #2 | 1 | 1.65 | 1.35 | 1.65 | 1.35 | NO EOW |
| | | 1.4 | 1.6 | 1.4 | 1.6 | MISS |
| SMIA #3 | 8 | 1.65 | 1.35 | 1.65 | 1.35 | NO EOW |
| | | 1.4 | 1.6 | 1.4 | 1.6 | MISS |
| Tx | 8 | | | | | |
| SMIA #1 | 3 | 1.65 | 1.35 | 1.65 | 1.35 | NO EOW |
| | | 1.4 | 1.6 | 1.4 | 1.6 | MISS |
| SMIA #2 | 1 | 1.65 | 1.35 | 1.65 | 1.35 | NO EOW |
| | | 1.4 | 1.6 | 1.4 | 1.6 | MISS |
| SMIA #3 | 6 | 1.65 | 1.35 | 1.65 | 1.35 | NO EOW |
| | | 1.4 | 1.6 | 1.4 | 1.6 | MISS |

TABLE VI. LIMIT TEST, DATA FIELD SYMMETRY

SMIA #1 = S/N 0002
SMIA #2 = S/N 0004
SMIA #3 = S/N 0008

| | STUB LOCATION | DATA FIELD SYMMETRY (NS) 1ST ½ FIELD | 2ND ½ FIELD | ERROR TYPE |
|---|---|---|---|---|
| Tx | 1 | | | |
| SMIA #1 | 7 | 640 | 360 | MISS |
| SMIA #2 | 6 | 250 | 750 | MISS |
| | | 600 | 400 | |
| SMIA #3 | 8 | 670 | 330 | MISS |
| | | 370 | 630 | |
| Tx | 6 | | | |
| SMIA #1 | 7 | 640 | 360 | MISS |
| | | 380 | 620 | |
| SMIA #2 | 1 | 250 | 750 | MISS |
| | | 600 | 400 | |
| SMIA #3 | 8 | 670 | 330 | MISS |
| | | 370 | 630 | |
| Tx | 8 | | | |
| SMIA #1 | 3 | 640 | 360 | MISS |
| | | 380 | 620 | |
| SMIA #2 | 1 | 250 | 750 | MISS |
| | | 600 | 400 | |
| SMIA #3 | 6 | 670 | 330 | MISS |
| | | 370 | 630 | |

TABLE VIII. LB-1 NEGATIVE TEST RESULTS

| TEST SUBSET | WORD GAP TIME μS | NEGATIVE TEST INPUTS | | | | RECEIVER STUB LOCATION | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | DROPPED BIT | EXTRA BIT | WRONG PARITY | INTERRUPTED TRANSMISSION | J2 | J4 | J6 | J8 | J10 | J12 | J14 | J16 |
| 1 | 3 | ✓ | | | | OK | OK | OK | OK | OK | OK | OK | OK |
| 1 | 3 | | ✓ | | | OK | OK | OK | OK | OK | OK | OK | OK |
| 1 | 3 | | | ✓ | | OK | OK | OK | OK | OK | OK | OK | OK |
| 1 | 3 | | | | ✓ | OK | OK | OK | OK | OK | OK | OK | OK |
| 1 | 1000 | ✓ | | | | OK | OK | OK | OK | OK | OK | OK | OK |
| 1 | 1000 | | ✓ | | | OK | OK | OK | OK | OK | OK | OK | OK |
| 1 | 1000 | | | ✓ | | OK | OK | OK | OK | OK | OK | OK | OK |
| 1 | 1000 | | | | ✓ | OK | OK | OK | OK | OK | OK | OK | OK |
| 2 | 3 | ✓ | | | | OK | OK | OK | OK | OK | OK | OK | OK |
| 2 | 3 | | ✓ | | | OK | OK | OK | OK | OK | OK | OK | OK |
| 2 | 3 | | | ✓ | | OK | OK | OK | OK | OK | OK | OK | OK |
| 2 | 3 | | | | ✓ | OK | OK | OK | OK | OK | OK | OK | OK |
| 2 | 1000 | ✓ | | | | OK | OK | OK | OK | OK | OK | OK | OK |
| 2 | 1000 | | ✓ | | | OK | OK | OK | OK | OK | OK | OK | OK |
| 2 | 1000 | | | ✓ | | OK | OK | OK | OK | OK | OK | OK | OK |
| 2 | 1000 | | | | ✓ | OK | OK | OK | OK | OK | OK | OK | OK |
| 3 | 3 | ✓ | | | | OK | OK | OK | OK | OK | OK | OK | OK |
| 3 | 3 | | ✓ | | | OK | OK | OK | OK | OK | OK | OK | OK |
| 3 | 3 | | | ✓ | | OK | OK | OK | OK | OK | OK | OK | OK |
| 3 | 3 | | | | ✓ | OK | OK | OK | OK | OK | OK | OK | OK |
| 3 | 1000 | ✓ | | | | OK | OK | OK | OK | OK | OK | OK | OK |
| 3 | 1000 | | ✓ | | | OK | OK | OK | OK | OK | OK | OK | OK |
| 3 | 1000 | | | ✓ | | OK | OK | OK | OK | OK | OK | OK | OK |
| 3 | 1000 | | | | ✓ | OK | OK | OK | OK | OK | OK | OK | OK |

TABLE VII. LIMIT TEST, 16 MHZ CLOCK FREQ.

| MASTER CLOCK | ERROR TYPE |
|---|---|
| MAXIMUM FREQUENCY | |
| 16.168 MHZ | MISS (NON-VALID MANCHESTER) |
| MINIMUM FREQUENCY | |
| 15.794 MHZ | MISS (BIT COUNT) |

CLOCK SOURCE TO TEST SET:

H & P 8015A PULSE GENERATOR

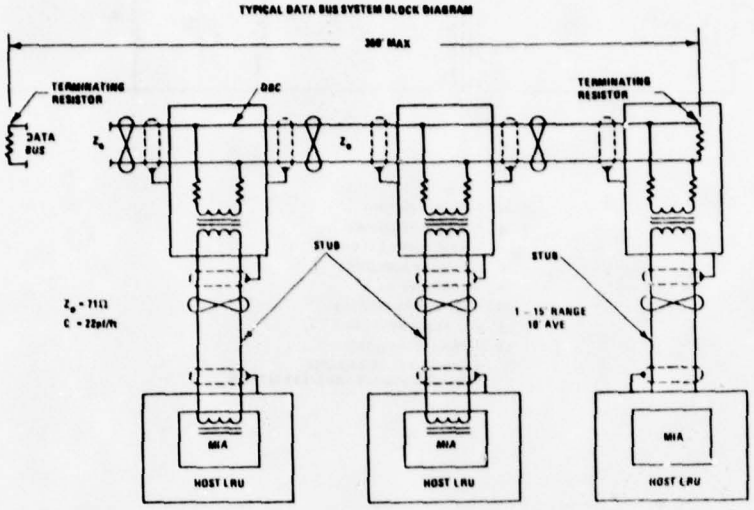
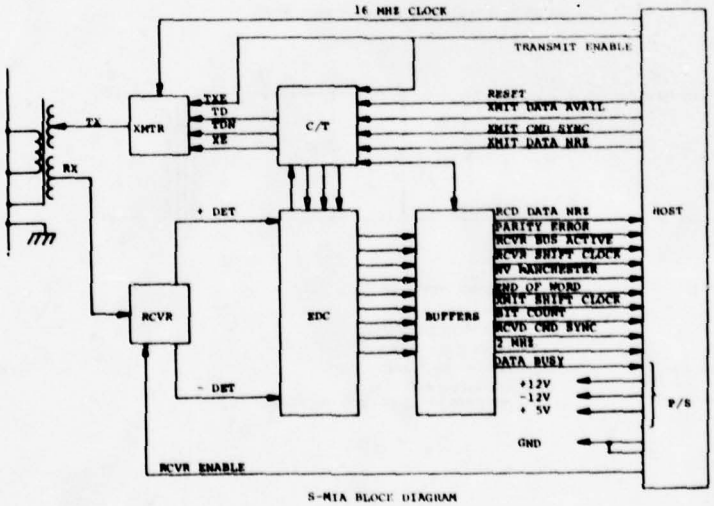


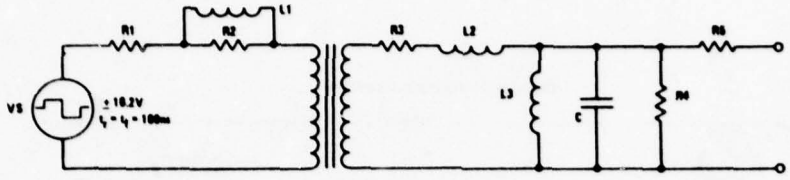

FIGURE 1




S-MIA BLOCK DIAGRAM

FIGURE 2

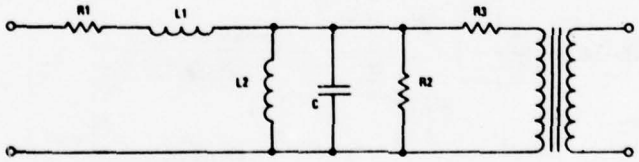
SMIA SOURCE IMPEDANCE (TRANSMIT MODE) MODEL




VS - SOURCE VOLTAGE
R1 - SOURCE RESISTANCE
R2, L1 - WAVE SHAPING NETWORK
R3 - PRIMARY RESISTANCE
R4 - CORE LOSS
R5 - SECONDARY RESISTANCE
L2 - LEAKAGE INDUCTANCE
L3 - PRIMARY INDUCTANCE
C - EQUIVALENT CAPACITANCE (WINDINGS, SHIELD AND STRAY EFFECTS)

FIGURE 3

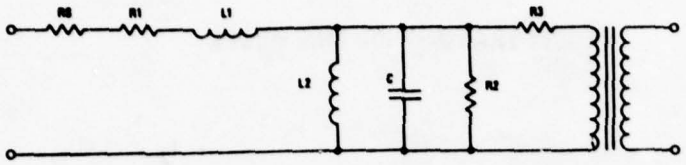
SMIA/OBIA INPUT TRANSFORMER (RECEIVE MODE) MODEL




R1 - PRIMARY RESISTANCE
R2 - CORE LOSS
R3 - SECONDARY RESISTANCE
L1 - LEAKAGE INDUCTANCE
L2 - PRIMARY INDUCTANCE
C - EQUIVALENT CAPACITANCE (WINDINGS, SHIELD AND STRAY EFFECTS)

FIGURE 4

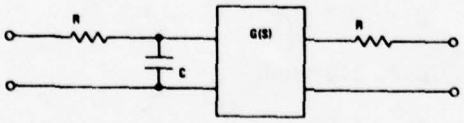
DATA BUS COUPLER (DBC) MODEL




RS = INTERNAL SERIES RESISTORS (50Ω) + 50Ω))
R1 = PRIMARY RESISTANCE
R2 = CORE LOSS
R3 = SECONDARY RESISTANCE
L1 = LEAKAGE INDUCTANCE
L2 = PRIMARY INDUCTANCE
C = EQUIVALENT CAPACITANCE
(INCLUDES WINDINGS, SHIELD AND, STRAY EFFECTS)

FIGURE 5

SMIA/DBIA INPUT FILTER MODEL

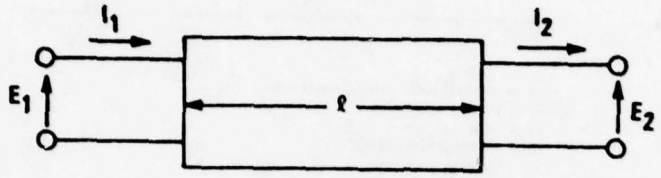



R1, C = FILTER INPUT IMPEDANCE
R2 = OUTPUT IMPEDANCE
G(S) = FILTER TRANSFER FUNCTION

$$G(S) = \frac{KS^2}{S^8 + b_7S^7 + b_6S^6 + \cdots + b_1S + b_0}$$

FIGURE 6

## LOSSY TRANSMISSION LINE MODEL




$$E_1 = E_2 \cosh \gamma\ell + I_2 Z_0 \sinh \gamma\ell$$

$$I_1 = E_2 \frac{\sinh \gamma\ell}{Z_0} + I_2 \cosh \gamma\ell$$

$$Z_0 = \sqrt{\frac{R + jwL}{G + jwC}}$$

$$\gamma = \sqrt{(R + jwL)(G + jwC)}$$

$C = 21.2$ pf/ft

$L = 116$ nHy/ft

$G = 9.6$ $\mu$mho/ft

$R = (.0024\sqrt{f} + 0.26)\ .03$ ohms/ft

FIGURE 9

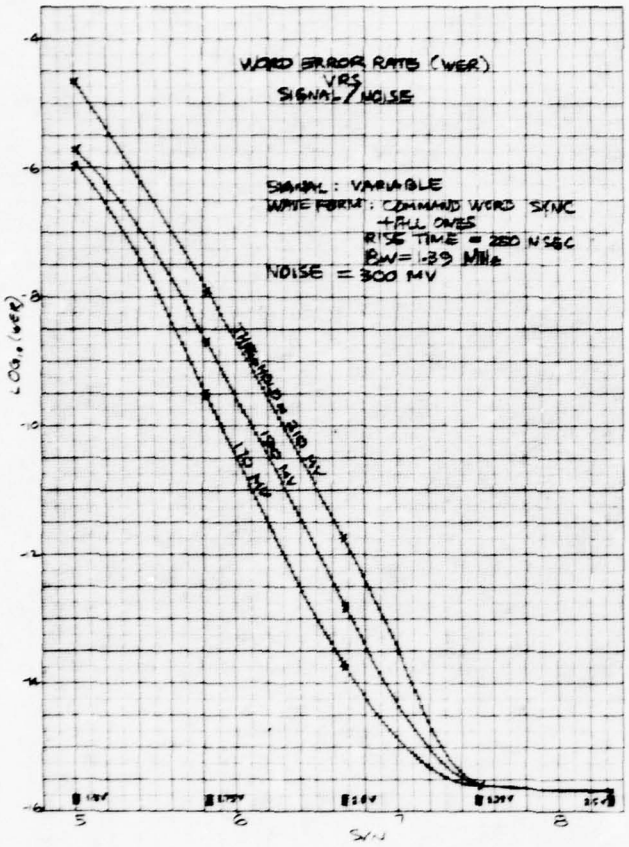



FIGURE 10

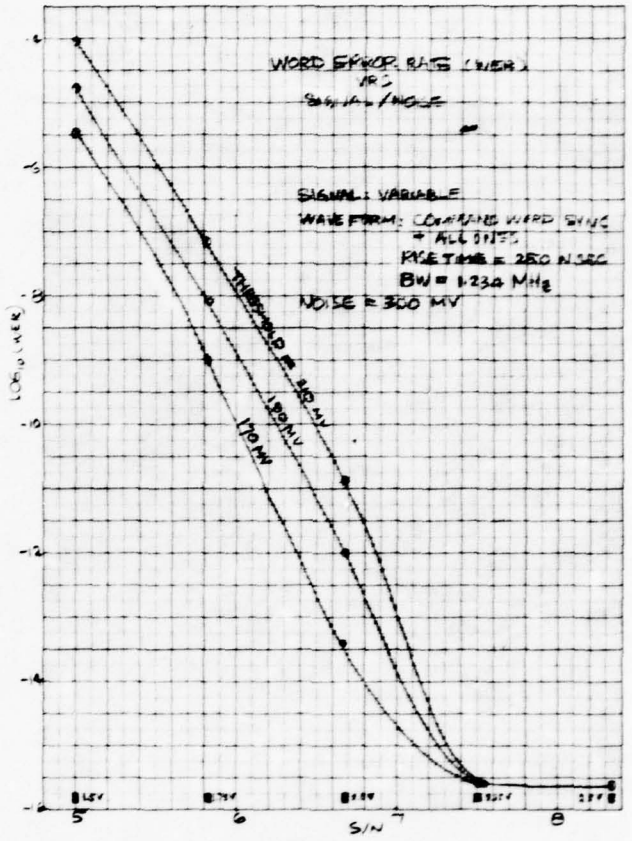



FIGURE 11

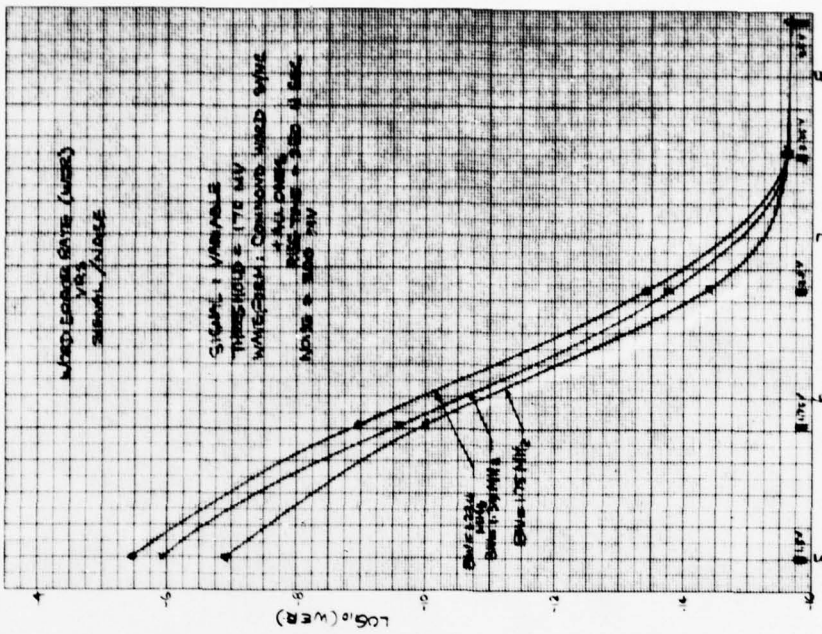



FIGURE 12

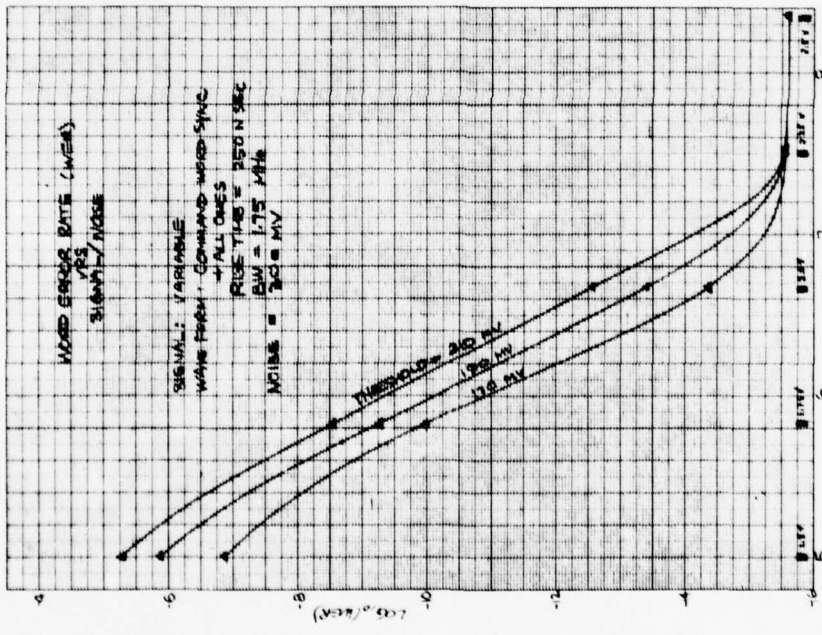



FIGURE 13

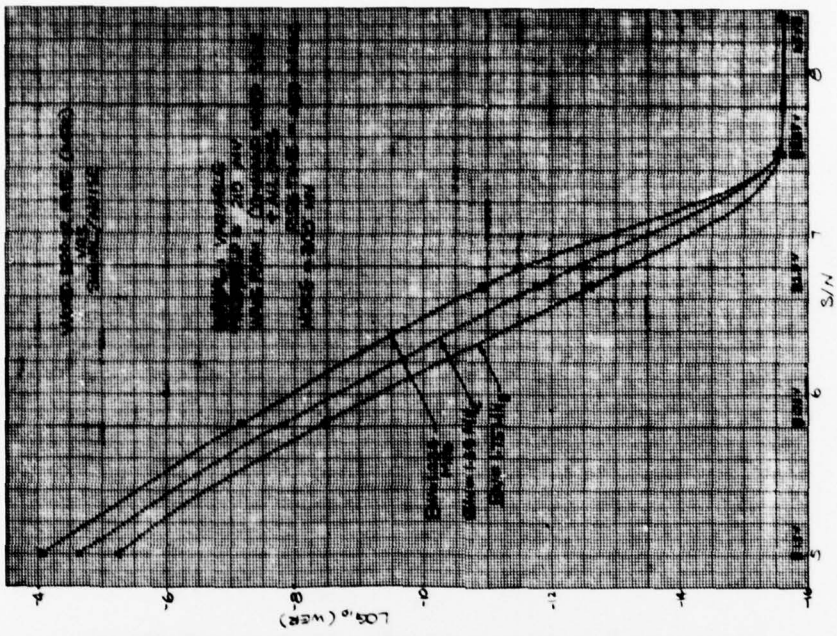


FIGURE 15

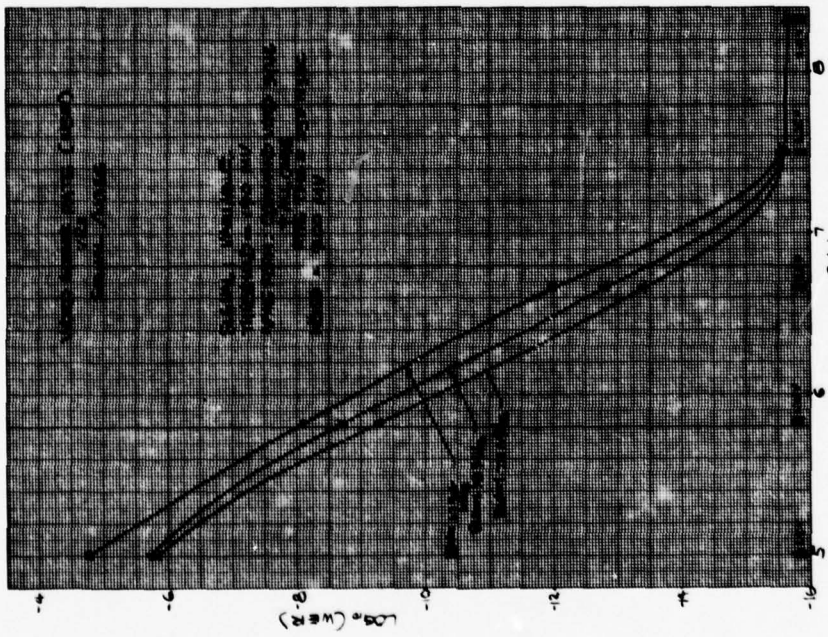


FIGURE 14

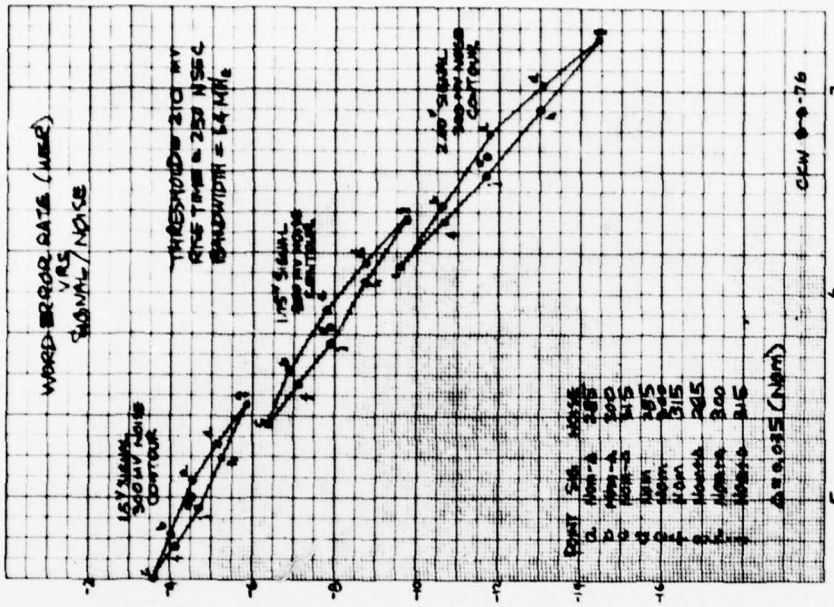



FIGURE 17

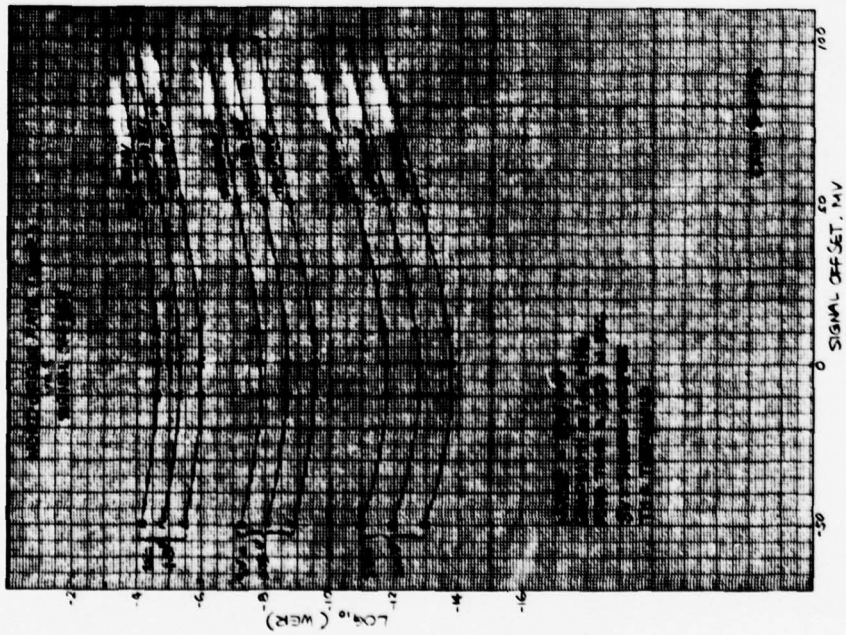



FIGURE 16

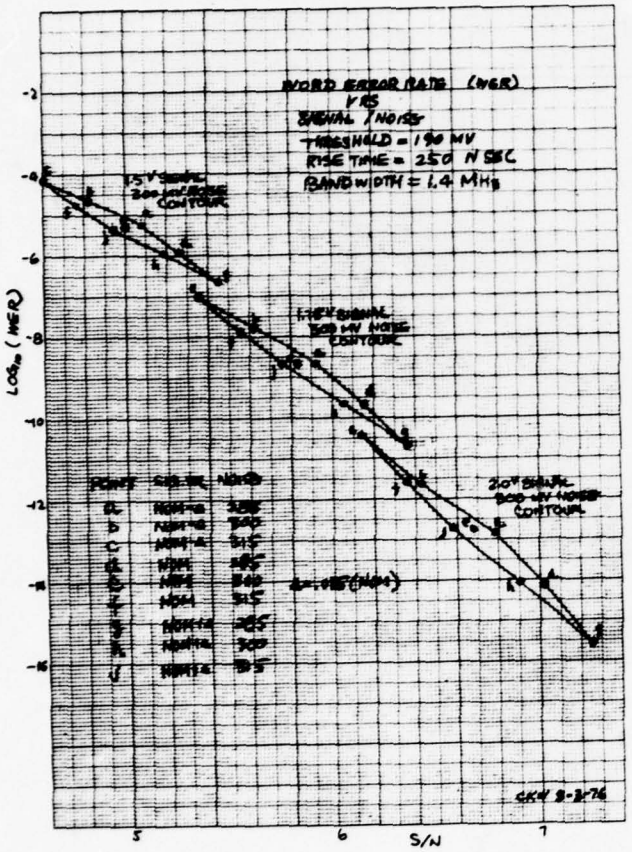



FIGURE 18

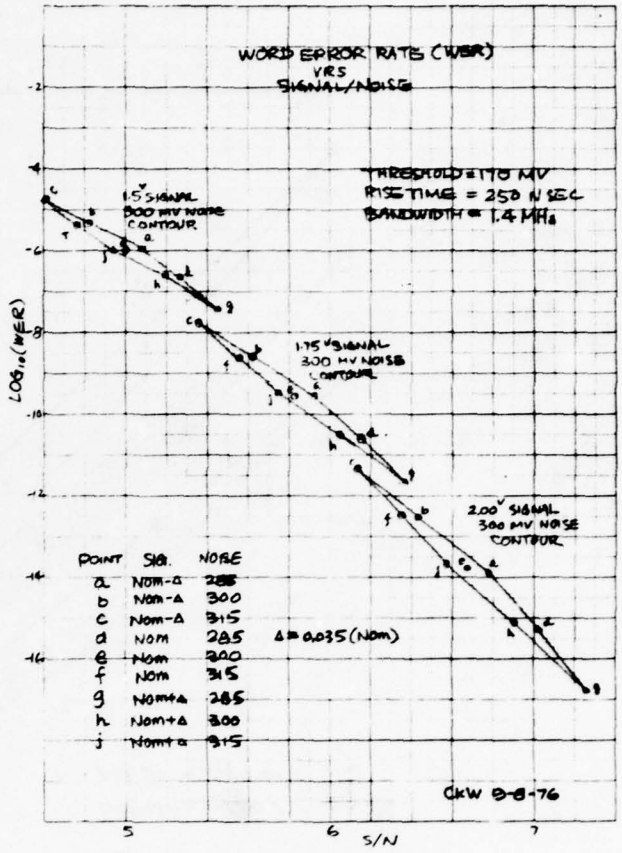



FIGURE 19

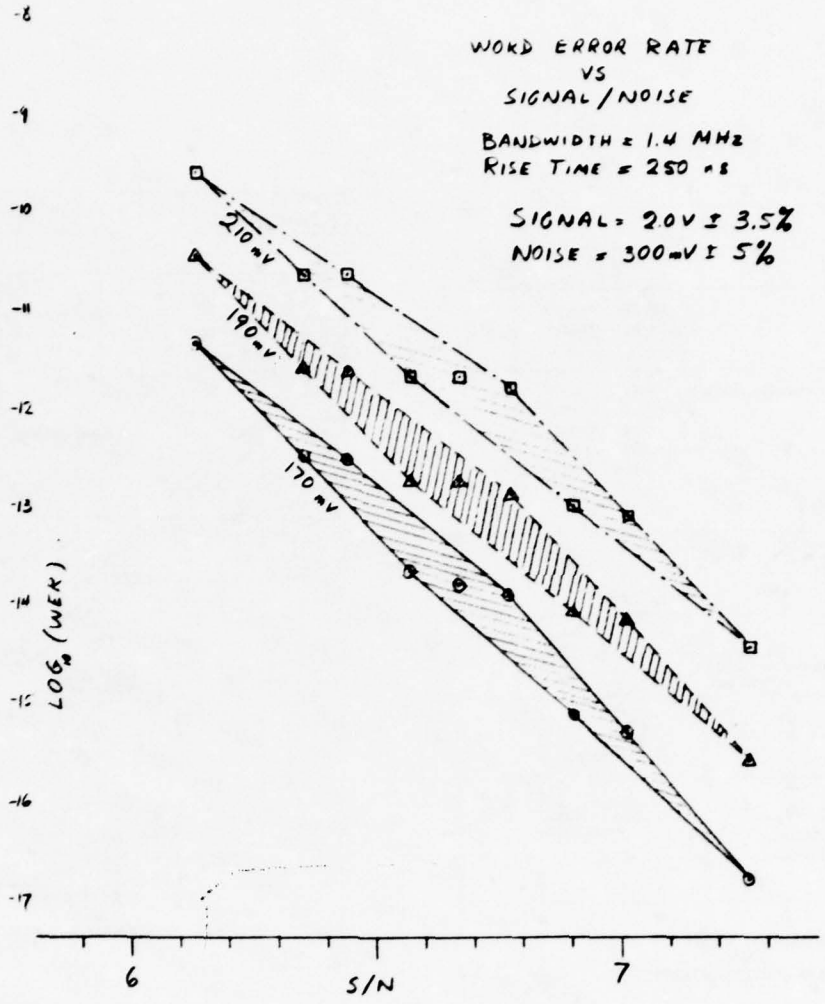

WORD ERROR RATE
VS
SIGNAL / NOISE
BANDWIDTH = 1.4 MHz
RISE TIME = 250 nS
SIGNAL = 2.0V ± 3.5%
NOISE = 300mV ± 5%
210 mV
190 mV
170 mV
LOG10 (WER)
-8
-9
-10
-11
-12
-13
-14
-15
-16
-17
6
S/N
7

FIGURE 20

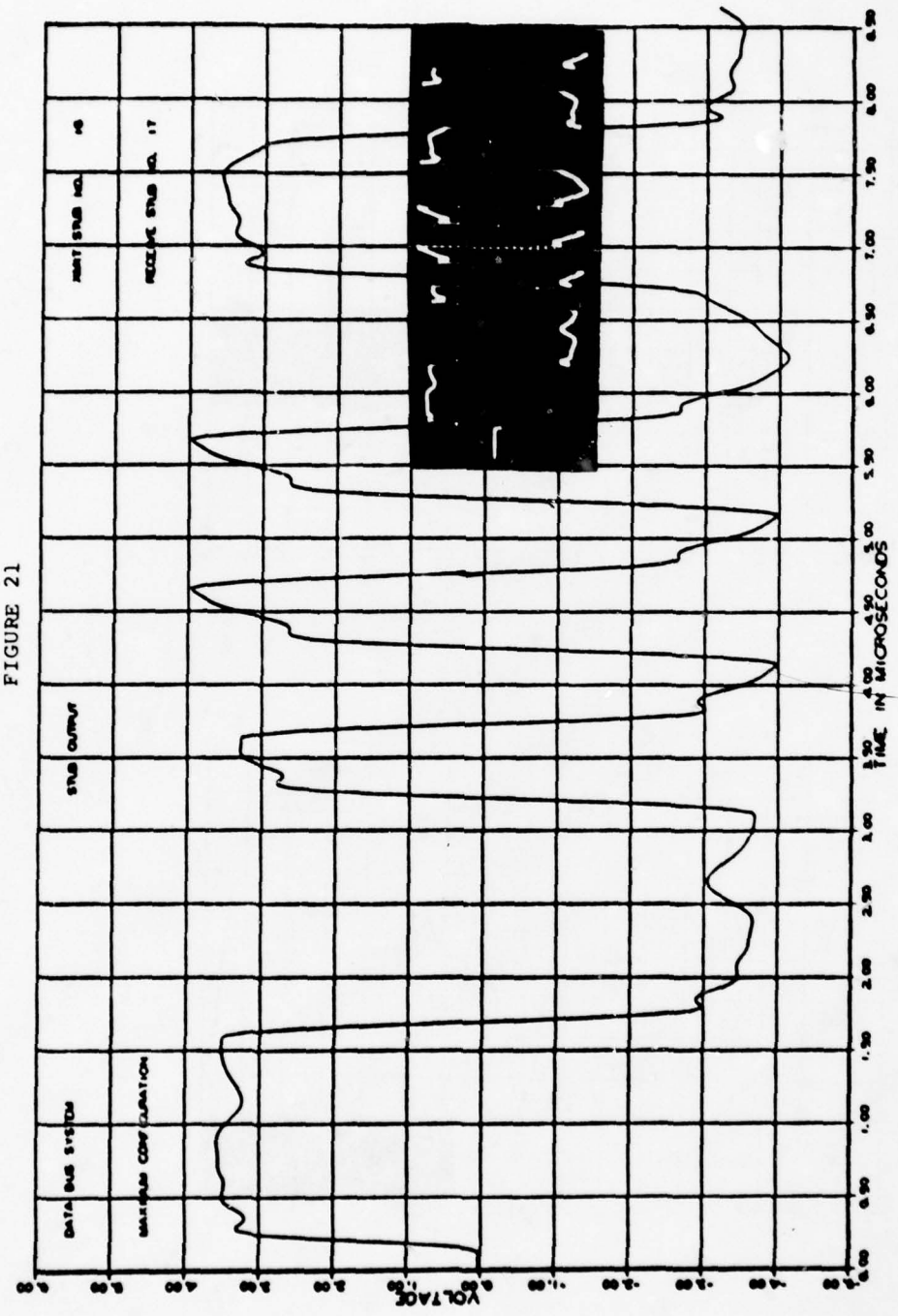

DATA BUS SYSTEM
MAXIMUM CONFIGURATION
STUB OUTPUT
XMIT STUB NO. 18
RECEIVE STUB NO. 17
VOLTAGE
TIME IN MICROSECONDS

FIGURE 21

FIGURE 22

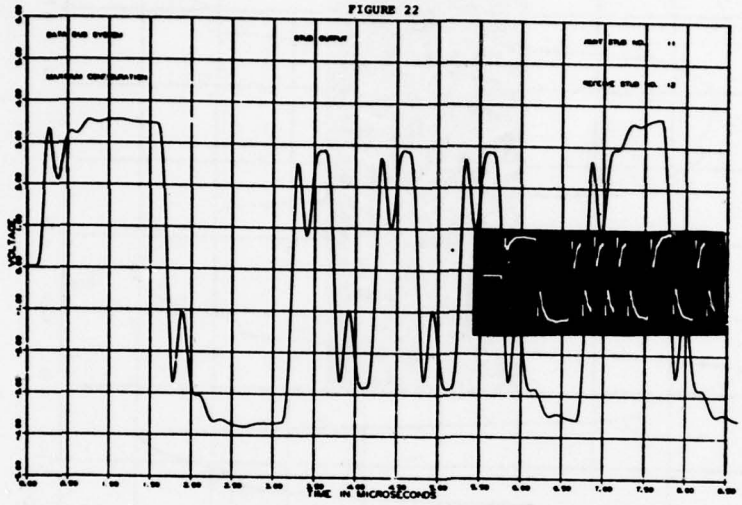
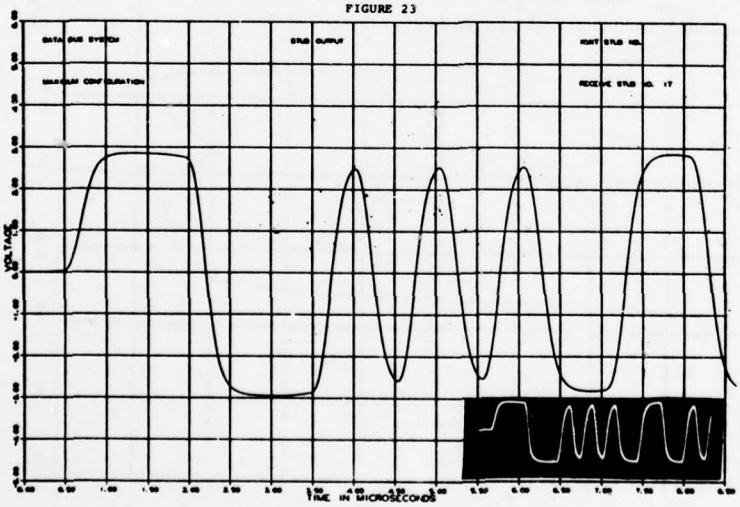


FIGURE 23



DATA BUS SYSTEM TEST CONFIGURATION

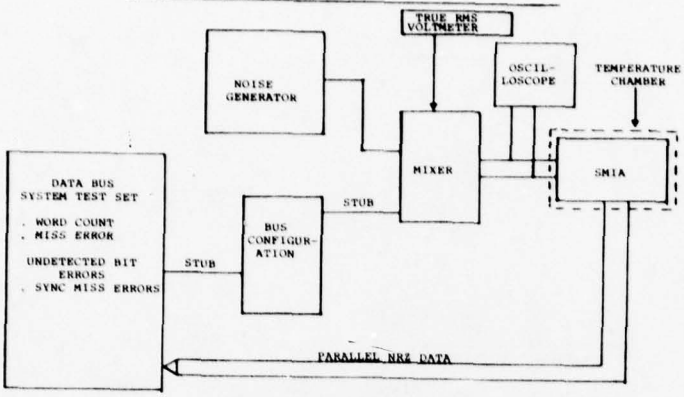



FIGURE 24

DATA BUS SYSTEM

ADJUSTABLE PARAMETER MIXER/AMPLIFIER UNIT

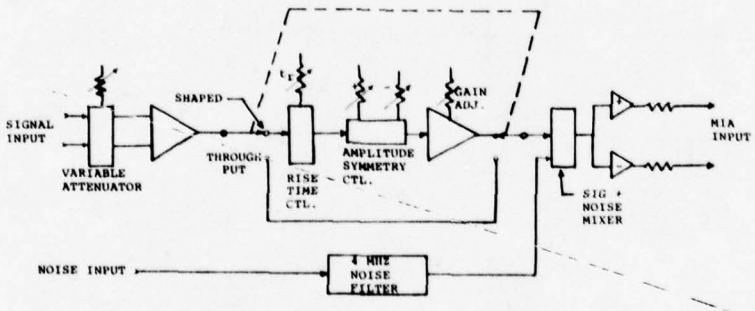



FIGURE 25

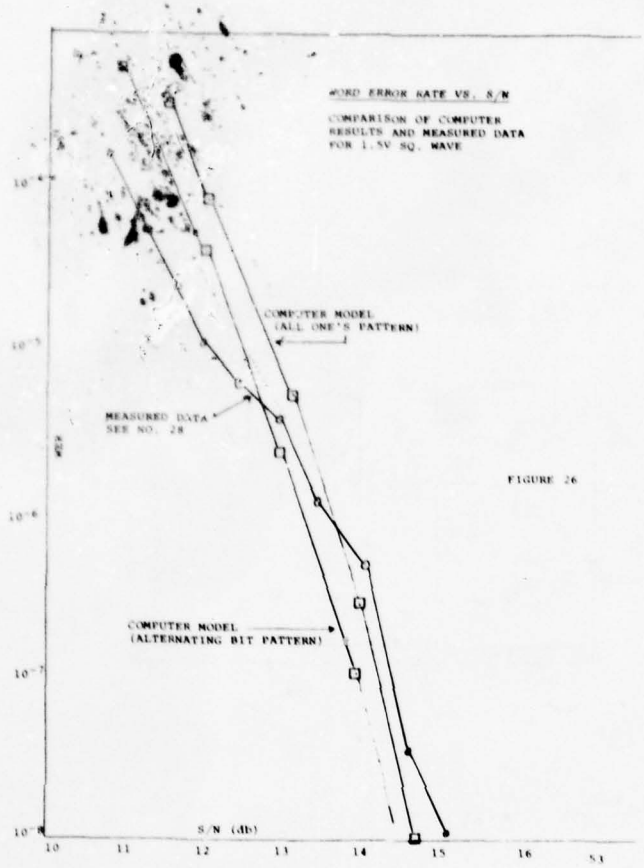
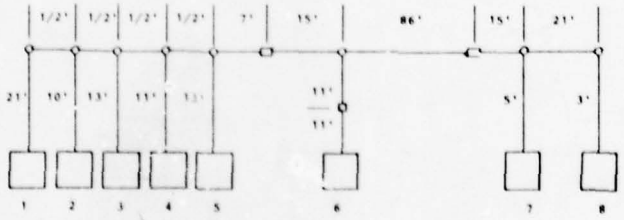



FIGURE 26

TEST CONFIGURATION (LB1)




STUB NUMBERS

FIGURE 26

# AIRCRAFT MULTIPLEX SYSTEM DESIGN AND INTEGRATION REQUIREMENTS FOR LOW NOISE DATA BUS OPERATION

By William S. Hermann and Dr. Clifford D. Skouby
McDonnell Aircraft Company, St. Louis, Mo.

## ABSTRACT

This paper describes the importance of noise reduction in operational aircraft multiplex systems. A number of noise sources such as intersymbol interference, white gaussian, impulse, common-mode, harmonic, and synchronization noise are discussed. In each case, methods of noise reduction are recommended. The effects of noise on bit error rate is illustrated for distortion-free as well as distortion-limited performance. Finally, techniques for evaluating data bus performance are described including undetected error rate verification and eye pattern measurements.

# AIRCRAFT MULTIPLEX SYSTEM DESIGN AND INTEGRATION REQUIREMENTS FOR LOW NOISE DATA BUS OPERATION

By William S. Hermann and Dr. Clifford D. Skouby
McDonnell Aircraft Company, St. Louis, Mo

## INTRODUCTION

This paper describes various system design and integration techniques for low noise bus operation in a typical fighter aircraft environment. Major sources of noise and signal distortion which reduce signal detection capability and increase bit-error rates will be discussed. Also techniques for noise reduction and signal distortion improvement are described for applications where stub effects, cable faults and non-linear lines must be tolerated. Since data bus characteristics can be minimized by control of noise, signal waveforms and coupling, the critical problems of installation can be minimized. For example, a non-linear bus can increase the level of intersymbol interference caused by overlapping of pulse tails in successive pulses. Also, wideband, distributed (Gaussian) noise and high amplitude impulse noise can be coupled into the data bus, particularly if common mode rejection is not effective. Harmonic noise generated by high rise-time transmitted waveforms are generally a problem with radiated signal effects coupled into high gain R.F. receivers external to the MUX system. They are also a source of intersymbol interference produced by the undamped overshoots in each bit resulting from excessive rise and fall times. In each of the descriptions of noise source, specific methods of noise reduction are recommended.

## SYSTEM REQUIREMENTS FOR LOW NOISE DATA BUS OPERATION

### NOISE SOURCES IN BASEBAND WIRE LINE MULTIPLEX

- INTERSYMBOL INTERFERENCE
- WHITE GAUSSIAN AND IMPULSE NOISE
- COMMON MODE NOISE
- HARMONIC NOISE
- SYNCHRONIZATION NOISE

### DATA BUS PERFORMANCE vs NOISE

- SNR vs BER
- EYE PATTERNS

GP78-0706-1

Finally, techniques for determining the influence of noise on system performance are described showing the relation of modulation type and noise level on bit error rate. Also, the use of eye patterns illustrate a technique for determining system margin against noise error at the best sampling point. Also, they provide an index of the sensitivity to timing error and permit the evaluation of error rate limits versus eye opening in a long train of random pulses.

INTERSYMBOL INTERFERENCE

Intersymbol interference is the most common source of distortion and increased error rate at megabit rates in wireline data bus systems. This waveform distortion is produced by the pulse-to-pulse phase/amplitude overlap of the pulse tail or decay transient of a preceeding pulse into a successive time slot. This effect is particularly a source of bit error at the end of each word where the pulse tail from the last bit is no longer overlapped by a following bit. Consequently, the pulse tail energy is sufficient to produce an additional bit detection, producing an invalid bit count.

## INTERSYMBOL INTERFERENCE

**DESCRIPTION**

- ADJACENT TIME SLOT OVERLAP OF PRECEDING PULSE TAIL WITH EXISTING PULSE

**SEVERITY**

- PRIMARY SOURCE OF DISTORTION/BIT ERROR

**CAUSE**

- PHASE/AMPLITUDE NON-LINEARITY OF BUS

**TEST CRITERIA**

- FOR ZERO INTERFERENCE (NYQUIST)
  - EQUAL ZERO CROSSINGS WITH "T" SECOND PERIOD PRODUCE CONSTANT BIT RATE OF 1/T SECONDS
  - EQUAL PERIODS BETWEEN TRANSITIONS

**SOLUTION**

- CONTROLLED TRANSMIT RISE/FALL TIME (TRANSIENT SUPPRESSION)
- CONTROLLED TRANSMIT DECAY (LINE COUPLING/DROOP)
- FILTERED RECEIVER INPUT WAVEFORM
- LIMITED OPEN-LINE FAILURE MODES
- LIMITED NUMBER/LENGTH OF LONG STUBS

GP78-0706-2

It should be noted that the magnitude of waveform distortion is important as it affects the value of each pulse at the sampling instant since bit detection and detection error are determined at this point. In "Data Transmission" by Bennett and Davey, the probability of bit error is determined for a polar, random, binary sequence in an additive white gaussian noise with optimum receiver filtering. The point in this case is that noise induced distortion is based on known, statistical distributions in order to define detection performance. This assumes a predictable influence of a uniformly distributed noise spectrum and constant amplitude in order to establish the influence on pulse values at the sampling instant. Conversely, Nyquist defined the waveform criteria for an "ideal channel" with zero interference as follows: First, the waveform zero crossings must be equally spaced in time having period of "T" seconds and a uniform bit rate of 1/T seconds. Secondly, the period between transitions must be equal, producing symmetry about the zero amplitude axis. This effect will be illustrated in later discussion of eye pattern tests.

One way to meet the Nyquist criteria is to filter and attenuate all previous pulse effects to zero. This filter response can be achieved by a sin X/X impulse response but is quite sensitive to sampling times and requires an ideal phase versus frequency characteristic. Consequently, it is better to examine the cause and effect produced by non-ideal but less complex solutions.

The basic cause of intersymbol interference is due to the non-linearity of the data bus versus frequency resulting in exponentially decaying transients occurring in a subsequent time slot. In other words, a purely resistive transmission line would have zero transient response. Nevertheless, an attempt to produce a resistive or zero reactance line is unrealistic in practical applications. Instead, a more practical solution is to avoid non-linear bus configurations insofar as practical plus control of the transmitted and received waveform distortion and thirdly, to minimize coupling of external noise. As a matter of priority, the bus linearity deserves major consideration since the effects of intersymbol interference or line induced channel distortion has a considerably greater impact on system performance than noise induced distortion.

Techniques to reduce bus non-linearity include minimal use of stubing, using terminal-to-terminal party line termination often referred to as a "daisy chain". Also, open circuit faults have a devastating effect on bus linearity particularly when occuring at the remote terminal connection to a long stub. A latter discussion of common-mode noise describes the effective use of grounded center taps to provide an impedance path to ground in the event an open occurs in one conductor of the bus.

Techniques to control waveform distortion include the use of sine wave line drivers to reduce harmonics injected on the transmission line and receiver filters to reduce distortion at the detector. Digital matched filters using active element CCD devices offer a possible solution, in conjunction with LSI microcircuit technology. Also a single chip LSI

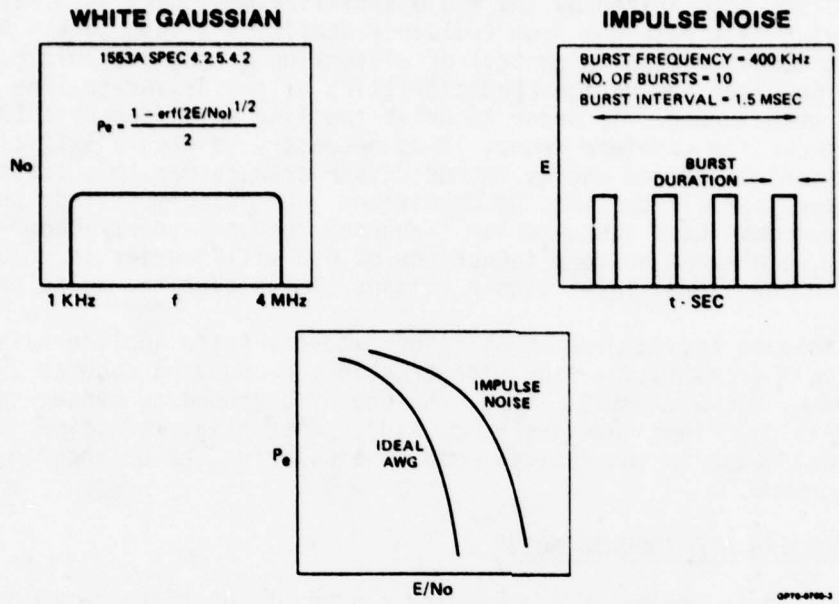
device has been developed to perform automatic digital equalization as described by Gerst and Diamond in "Proc IRE" Vol 53, July 1961. This technique was popularized by the audio amplifier designers as a method of achieving flat response from frequency sensitive transducers. An important consideration in control of distortion of the transmitted waveform involves the design characteristics of the driver-to-line coupling transformer. In order to drive the line specified by 1553 without excessive waveform droop, it is necessary to have a sufficient time constant or stored energy in the driver transformer to sustain the signal level for all bus loading conditions. Or, simply stated, the transformer must have adequate low frequency response under load. Consequently, a minimum primary inductance of 6.3 millihenries is recommended to sustain the signal level within voltage limits over the pulse period.

Techniques to minimize coupling of external noise include balanced, twisted pair transmission line with transformer coupling such as described by MIL-STD-1553B. Also, the use of a grounded, center-tapped coupling as described previously, provides substantial reduction in common mode noise in non-faulted mode by equalizing the balance in either line to ground.

## WHITE GAUSSIAN AND IMPULSE NOISE

Line induced noise is a secondary source of interference which has considerably less impact on detection error rates. This, of course, assumes that reasonable precautions are taken such as shielding, loading, grounding, decoupling and filtering. In addition, 1553 calls out transformer line coupling and balanced, shielded, twisted-pair transmission line as well as biphase signaling to minimize the influence of external noise on data bus performance. Assuming that such precautions have been taken, it is nevertheless desirable to predict the probability of detection error as related to expected levels of signal and noise. Since detection probability or error probability is related to the noise power within a specific bandwidth required for signaling, it is convenient to specify average noise power with a bandwidth equal to the data rate. White gaussian noise is particularly useful as a standard definition in which the total average noise power is proporational to bandwidth. It should be noted that noise in the Nyquist bandwidth, is also frequently used to define a noise power of -3dB less than noise defined in the bit rate bandwidth. Gaussian noise provides a convenient means of defining a density and distribution for noise spectrums within the detection bandwidth. As previously described, detection thresholds occur at the sampling point when the signal value exceeds the noise level by a predetermined value. This reference can be defined in a number of comparisons such as average signal power for a random pulse train or alternately for a given signaling sequence. Since this signal to noise relationship is essential to determining error probability at the sampling instant, specific conditions must be established. In Bennett and Davey on "Data Transmission", this analysis was determined for a polar, random, binary sequence with raised-cosine waveform with optimum receiver filtering

## COMPARISON OF GAUSSIAN vs IMPULSE NOISE




and additive white gaussian noise. *The probability of error for this condition is:*

$$P_e = \frac{1 - erf\ (2\ E/No)^{1/2}}{2}$$

where E is the average signal output power received and No is the average noise power in the bandwidth equal to the signaling frequency. This detector performance in a given signal-to-noise environment requires an excess signal above the detection threshold at the sampling instant as previously discussed. The gaussian noise distribution permits statistical prediction of power levels at the instantaneous point of sampling. Nevertheless gaussian noise is ideal in the sense that both amplitude peaks, spectral bandwidth and energy distribution are not representative of typical aircraft interference. From a practical point of view, impulse noise having either random or periodic impulse duration, frequency of occurance and burst interval more typically represent chattering relays, or switching currents in data bus subscriber pairs. Relay noise is generally characteristic of a short burst interval of a few miliseconds duration or less in periodic or randomly spaced groups. In order to analytically model this type noise, it is necessary to have a statistical description of the pulse amplitudes, pulse durations and pulse time occurances and distribution. The solution will typically take the form

of Poisson shot noise model in which the random area (a i) of an impulse occurs at the random time instant ti in any observation interval of To seconds.

$$N(t) = \sum_{i=-\infty}^{\infty} a_i \, \delta \, (t - ti)$$

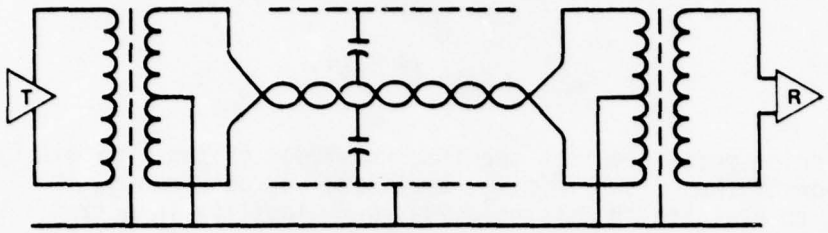
Since the noise model requires specific knowledge of the probability of distribution of the pulses in time, amplitude and occurrence, it is more convenient to evaluate the system noise susceptability in a test laboratory. Consequently, a typical transformer coupled data bus with shielded, twisted pair transmission line was evaluated in the presence of impulse noise with various bus grounding and shielding configurations. These test results are illustrated in the next paragraph on common mode noise.

COMMON MODE NOISE

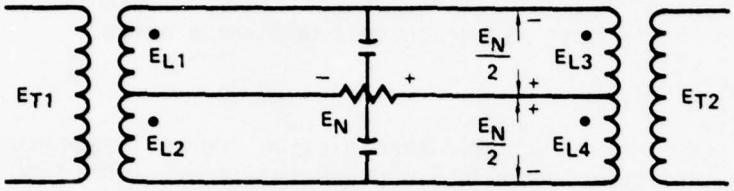
The benefits of balanced pair transmission are well recognized as a means of reducing common mode noise prevalent in unbalanced or ground return type transmission lines. Consequently, MIL-STD-1553 has specified balanced, twisted pair transmission lines in order to reduce the effects of common-mode noise coupling. This noise is coupled into a line to ground signal due to noise currents developing a voltage across finite resistance paths to ground which appear in series with the legitimate signal to ground path. In contrast, methods of controlling common mode noise in a balanced line are not as well recognized. One argument is that no ground reference exists in a balanced line, consequently the problem is non-existent. A second argument is that use of a single ground reference does not permit a ground circulating current. Both of these viewpoints fail to recognize that: first, there is a capacitive coupling to ground from each signal line in the balanced pair to the shield which is grounded to avoid electrostatic coupling. Secondly, the ground coupling is not to a single point since the capacitive effects are distributed over a wide variety of ground points therefore circulating currents will be produced. Also, the total capacitive effects from each line to ground are not equal consequently voltage unbalance and circulating currents to ground occur which increase common mode noise in the balanced signal.

To minimize this noise, it is necessary to minimize the unbalance. That is to say, if the unbalance to ground is zero, then the common mode coupling is zero. To accomplish this in a relatively simple manner, the centertap of the line driver transformer and each line receiver transformer should be grounded as illustrated. It is important to emphasize that the centertap balance can be controlled to relatively high degree of accuracy within fractions of a percent unbalance. In contrast, the nominal line to shield capacitance of 47 picofarads per foot can vary as much as ten to twenty percent from each line to shield producing relatively high levels of unbalance and common mode noise. The simplified circuit illustrates the necessity for grounding the centertaps at

## BALANCED LINE COUPLING




**SIMPLIFIED CIRCUIT - GROUNDED CT**




$E_{L1} = E_{L2} = E_{L3} = E_{L4}$ FOR IDEAL CENTER TAP

$E_{L1} - E_{L3} = E_{L2} - E_{L4} \cong 0$ WITHIN CT BALANCE ERROR

$E_N$ = INSTANTANEOUS COMMON MODE NOISE VOLTAGE DEVELOPED ACROSS THE GROUND PATH RESISTANCE

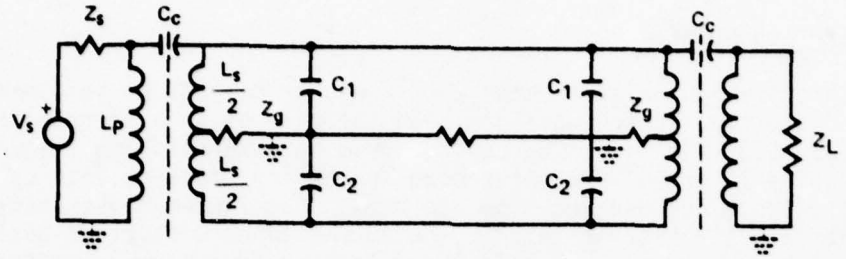
GP78-0708-4

both ends of the bus. If a perfect balance occurs, then $E_{L1}$, $E_{L2}$, $E_{L3}$, and $E_{L4}$ are equal consequently the centertap potential of both transformers are equal and zero common mode voltage is developed in the coupling transformers. If either zero or one centertap is grounded, the capacitance to ground produces a considerably higher unbalance and considerable increase in common mode noise coupling results.

Several additional observations can be made from the grounded centertap circuit. First, the ground path carries only the unbalance current and not the signal current. Consequently, the integrity of the ground, or the resistance of the path does not have a significant effect on the signal or the common mode rejection capability. Secondly, if a catastrophic failure or open occurs in one line, the second line will operate as an unbalanced, line-to-ground, transmission line, with additional common mode noise but with a total functional capability.

It is recognized that the simplified equivalent circuit does not include all the actual circuit elements such as transformer winding resistance, winding capacitance, leakage capacitance, mutual coupling, transmission line resistance, etc. As can be seen from the illustration, the detailed equivalent circuit adds considerably to the task of circuit analysis. Also the additional detail has a negligible influence on the effects of unbalance on common mode noise rejection with the exception that capacitive coupling from primary to secondary (line to terminal) provides an unbalance to ground if the terminal is an unbalanced, grounded configuration. To minimize the primary to secondary capacitive coupling, the use of an electrostatic shield is strongly recommended.

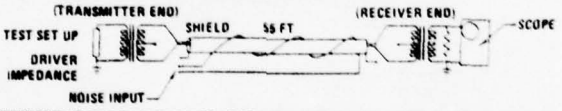
# LUMPED CIRCUIT EQUIVALENT BALANCED LINE DATA BUS




The results of laboratory tests verify the postulated performance of a balanced pair transmission line to reduce common mode noise using grounded, center tapped coupling for both the transmitter and receiver. In the illustrated test, a 55 foot shielded twisted pair was inductively coupled to an impulse noise source of 60 volts and 0.5 microsecond pulse-width. As can be seen from the waveforms to the lower left of the illustration, a peak noise signal of 3.2, 2.0, and 2.8 volts was induced

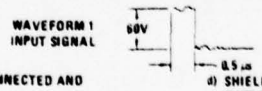
## MULTIPLEX TRANSMISSION LINE TESTS TWISTED PAIR, BALANCED LINE

**PRELIMINARY NOISE INTERFERENCE STUDY - LINE PICKUP**

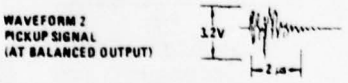



STANDARD AIRCRAFT TWISTED SHIELDED PAIR, 22 GAUGE, 55 FT LONG CABLED WITH A SINGLE UNSHIELDED WIRE CARRYING NOISE SIGNALS.

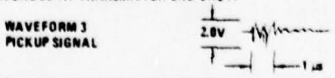
**TEST 1. SINGLE WIRE CARRYING POSITIVE PULSE 0.5 MICROSECOND WIDE, 60 VOLTS PEAK AMPLITUDE.**




a) TRANSMISSION LINE SHIELD NOT CONNECTED AND TRANSFORMER CENTER TAPS OPEN.




b) SHIELD NOT CONNECTED AND TRANSFORMER CENTER TAP *GROUNDED AT TRANSMITTER END ONLY.*



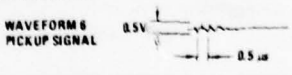

c) SHIELD NOT CONNECTED AND TRANSFORMER CENTER TAP GROUNDED AT RECEIVER END ONLY.



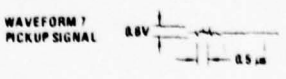

d) SHIELD NOT CONNECTED AND TRANSFORMER CENTER TAPS GROUNDED (BOTH ENDS)




e) SHIELD GROUNDED AT TRANSMITTER END AND CENTER *TAPS GROUNDED.*




f) SHIELD GROUNDED AT BOTH ENDS AND CENTER TAPS GROUNDED.

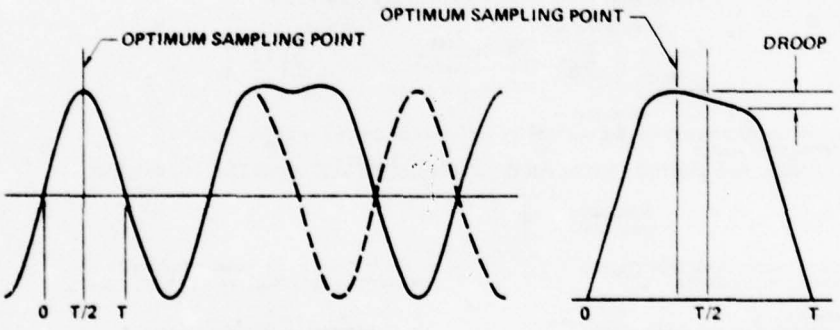


while minimizing the steep wavefront condition. Some caution should be observed in control of detector sampling time; however, to avoid reduction in signal margin above threshold, as described in the next paragraph on synchronization.

SYNCHRONIZATION NOISE

Synchronization noise creates zero crossing deviations in timing due to phase distortion in the waveform. In the case of an asynchronous data bus such as 1553, timing can be derived from an internal high frequency clock, usually 12 or 16 MHz, referenced to the invalid sync code or from a reconstructed clock derived from the data. For random noise occurance, the probability of error is less in the former since distortion must occur in the 3 bit sync code interval in order to produce timing deviation. The 1553 standard permits a maximum zero crossing deviation of 25 nanoseconds or five percent of the bit interval. This timing error will cause deviation from the optimum sampling point irrespective of other distortion in the bit waveform. The shift from optimum sampling will result in reduced detection margin above threshold for both sinusoidal and trapezoidal waveforms, assuming a permissable ten percent droop in the latter case. For the sinusoidal case, the amplitude reduction to 99.69 percent of optimum results with a zero crossing deviation of 25 nanoseconds. For the trapezoidal case, the amplitude reduction is 0.99667 percent of optimum with 25 nanoseconds crossing deviation, 100 nanoseconds rise/fall time and 10 percent droop. In both cases, this represents a 0.05 dB loss in signal-to-noise ratio and negligible change in bit error rate of the system.

## OPTIMUM SAMPLING/SYNCHRONIZATION




*SINE WAVE OUTPUT*

*TRAPEZOID OUTPUT*
(100 n SEC RISE/FALL TIME, 10% DROOP)

AMPLITUDE REDUCTION FROM UNITY WITH 25 NANOSECOND DEVIATION FROM MID-POINT SAMPLING

$$\text{SIN}\left|\frac{(500-25)}{500}\times 90^{\circ}\right| = \text{SIN } 85.5^{\circ} = 0.9969$$

$$1-\frac{25}{1000-200}\times\frac{10\%}{100} = 0.99667$$

corresponding respectively to the first test with ungrounded shield and open centertaps on both coupling transformers, the second test with ungrounded shield and only the transmitter centertap grounded, and thirdly, the ungrounded shield and only the receiver centertap grounded. It should be noted that in this group of tests, the noise pickup was relatively high, i.e., in excess of 2 volts and in no case were both centertaps grounded. In the second group of tests, illustrated on the lower right, a peak noise signal of 0.4, 0.5, and 0.6 volts was induced corresponding respectively to test number four with both centertaps grounded and shield not connected, test number five with both centertaps grounded and shield grounded at transmitter, and test number six with both centertaps grounded and shield grounded at both ends. It should be noted that the noise pickup voltages are relatively low in this group of tests, i.e., approximately 0.5 volts and in all cases both centertaps were grounded. In other words, the centertaps are effective in reducing common mode noise by a factor of six to one when grounded at both the transmitter and receiver.

HARMONIC NOISE

McDonnell Aircraft specifications for line driver waveform for the F-15 aircraft (MCAIR specification H-009) and for the F-18 aircraft (MCAIR specification A-38188) limit the harmonic output of the line driver to essentially a sine wave. This is in contrast to a number of system integrators who permit a square wave or more accurately a trapezoid with limited rise time and limited droop. It is well to use a reference to "Data Transmission" by Bennett and Davey which states that except for the waste of transmitted power, it is immaterial what the curve for the transmitting filter (waveform harmonic content) is like for frequencies above the bit rate if the receiving filter has no response at these frequencies. This criteria is intended to apply only to multiplex receiver response and disregards other receivers on the aircraft. Since a number of communications, navigation, IFF and ECM receivers have sensitivities of one or two microvolts into 50 ohms input impedance, the harmonics of a 10 volt (zero-to-peak) data bus driver output can represent a relatively high power interference within the passband. Admittedly, the radiated harmonics of the data bus driver can be limited by shielding nevertheless suppression of interference begins with reduction of transmitted harmonics where practical. MCAIR has produced more than 350 F-15 aircraft to date and experienced no difficulty with data bus radiation problems. We believe that sine wave driver outputs are a commonsense approach to radiation reduction. This requires a more costly linear amplifier driver but represents a good investment in RFI suppression.

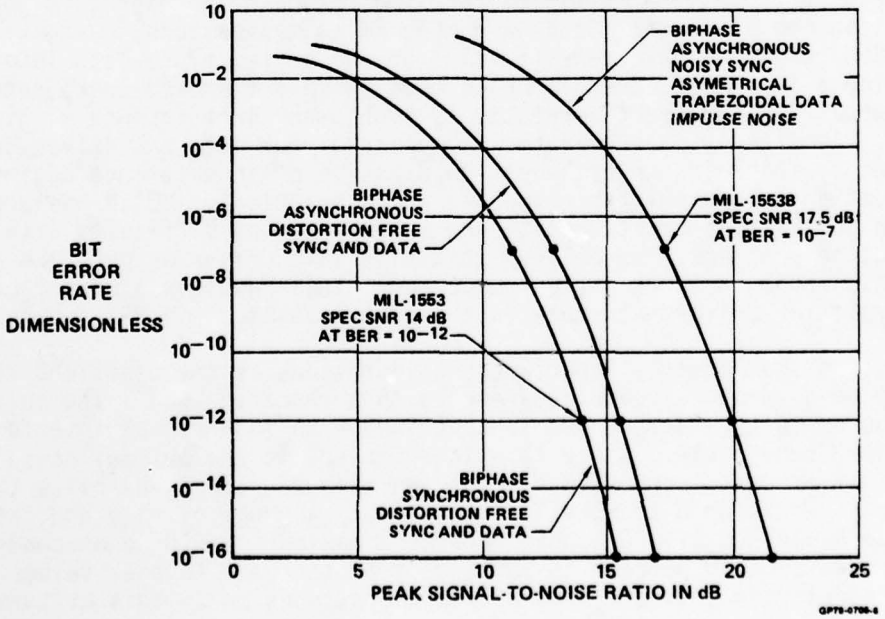
A second, but equally important consideration, is the transient response of a sine wave versus a quasi square wave to a reactive load. The pulse tail produced in the latter case is contributor to intersymbol interference as described previously. Since this interference is the primary cause of detector bit errors, every effort should be made to reduce the pulse tail distortion. Revision B to MIL-STD-1553 permits a range of rise and fall times from a minimum of 100 nanoseconds to a maximum of 300 nanoseconds when measured from 10 percent to 90 percent of the peak to peak value. The sinusoidal rise and fall time of 295 nanoseconds meets this criteria

A second cause of deviation from mid-point sampling is not related to noise effects but nevertheless produces a similar shift in sampling point. Consequently it is a source of reduction in amplitude at the sampling point and hence loss of detection margin. This deviation in sampling is caused by the resolution or granularity of timing produced by the 12 or 16 MHz clock. Since the leading edge of a clock pulse which is most nearly coincident with the mid-point of the bit period is used for timing, a finite shift in sampling point results from the random difference in coincidence. Obviously, a 12 MHz clock will have a more course granularity and consequently a greater reduction in detection margin. Since coincidence is limited to one half of one clock period maximum, the 12 MHz clock could have a lack of mid-point coincidence by as much as 40 nanoseconds. This worst case clock deviation is the same magnitude of error as in the zero crossing deviation error and hence has a relatively minor influence on bus performance.

## BIT ERROR RATE VS SIGNAL TO NOISE RATIO

In the previous discussion of gaussian and impulse noise, the computation of bit error probability versus signal-to-noise ratio (SNR) was shown in a non-quantitative example. In this illustration, the actual effect of SNR on system performance in terms of bit-error-rate (BER) is shown. The curve on the left shows the BER versus SNR for distortion-free, "synchronous", biphase signaling while the center curve shows a slightly reduced performance for distortion-free, "asynchronous", biphase signaling. The curve on the right shows the performance with a 6dB degradation from the

**PERFORMANCE COMPARISON OF DISTORTION-FREE vs PRACTICAL BIPHASE SIGNALING**

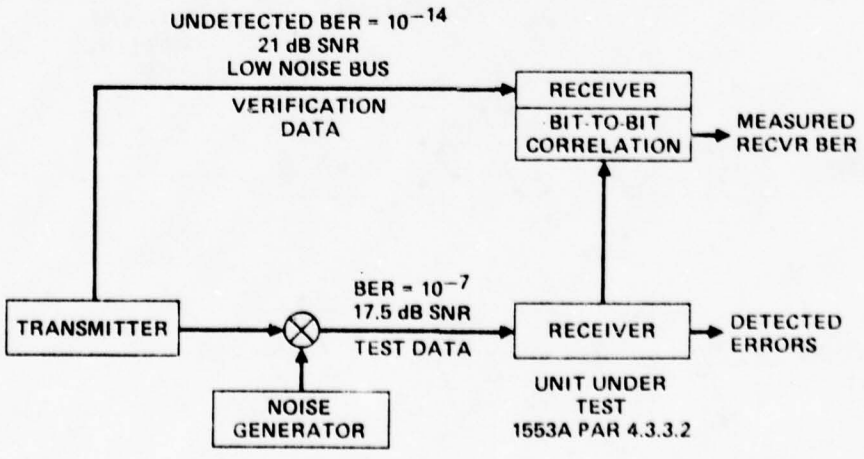


distortion-free, asynchronous detector to allow a practical level of waveform distortion such as trapezoidal waveshape with droop, non-symmetrical zero crossing and sync distortion. Other noise effects are analytically compensated for as a function of SNR with the exception that the distortion-free performance is based on gaussian noise. In contrast the practical curve includes a margin for the effects of impulse noise.

It should be noted that the "B" revision (1553) performance requirement of $10^{-7}$ BER at a 17.5 dB SNR appears on the distortion-limited performance curve. This value of SNR was intentionally selected to produce a corresponding value of BER which is sufficiently high ($10^{-7}$) to permit test and performance verification within a reasonable time period. In contrast, the original 1553 performance requirement of $10^{-12}$ BER at 14dB required considerable test time to verify. By adding more noise, the detector performance can be determined at a higher BER point on the curve with substantially less test time required.

The verification of detector performance in terms of BER should consider the measurement of both detected and undetected errors. To measure undetected errors which do not correlate with the transmitted signal and are not detected by the terminal under test, it is necessary to compare the transmitted and received signals on a bit-to-bit basis. It is also important to have an accurate reference of the transmitted data for comparison with the detected data from the terminal under test. To achieve a highly accurate transmitted data reference, the data bus should be a low noise type with typical SNR of 21 dB. This is generally not a problem, since the bus length and linearity can be controlled in a laboratory test configuration. Since this transmitted data reference has a theoritical error rate of $10^{-14}$ at 21 dB, it becomes an infinitely accurate reference to the $10^{-7}$ error rate system under test. To verify detector performance, the noise output mixed with the test signal should be increased to produce 17.5 dB SNR and the correlated transmit/receive error-rate determined.

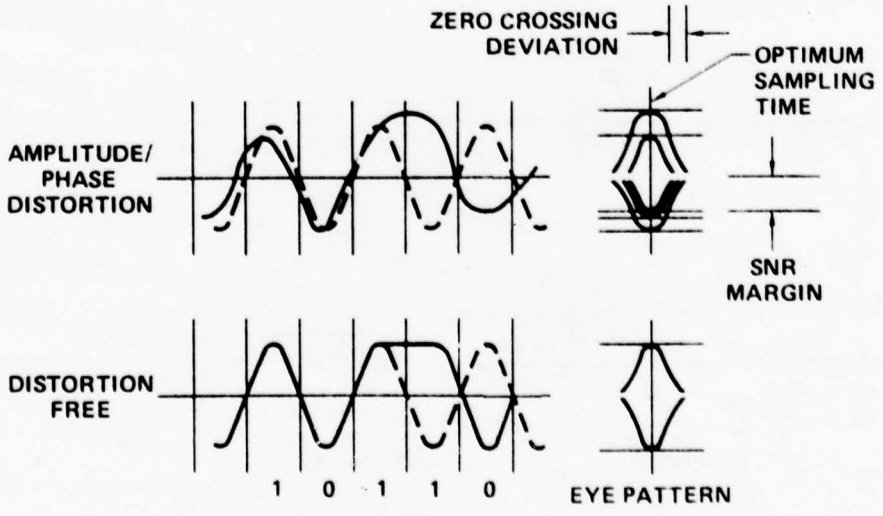
## UNDETECTED ERROR VERIFICATION



## EYE PATTERNS

Eye patterns, so-called for their resemblance to a human eye, are a useful laboratory method of evaluating random interference effects in a long pulse train. By observing an oscilloscope, with horizontal sweep externally synchronized to the bit rate, the bits can be superimposed over the same sweep interval. This overlapping effect, allows the observer to evaluate intersymbol interference in terms of the vertical and horizontal eye openings. For example, the vertical opening corresponds to the sampling point; and the horizontal opening, to the transition in the bit period. Consequently, variations in the horizontal pattern result from variation in bit transition time. A perfect eye pattern, corresponding to a distortion free signal, has maximum eye opening and conforms to the Nyquist criteria for zero intersymbol interference described earlier. The ratio of the measured eye opening, produced by the overlapping patterns with random distortion, as compared to the maximum opening, indicates the limits in waveform distortion. Also the patterns are quite useful in determining the optinum sampling point where the eye is open widest and in observing the effects of zero crossing deviation. As pointed out by Lucky, Salz and Weldon in "Principles of Data Communications", the sensitivity of the system to timing can be observed as the sampling time is varied and the signal distortion limits appear as a vertical width or closure at the sampling point. As this text further points out, this degree of opening can be used to bound the probability of error if the noise distribution is known. Typically, an eye opening of 0.5 might be used as measure of reliable data transmission. Nevertheless it is not a direct measure of bit-error-rate as described previously and should not be considered a substitute for BER verification tests. It is however useful in determing the minimum margin in a given data sequence, particularly a random sequence, where the varying overlap can be constantly monitored.

## EYE PATTERN EFFECTS vs DISTORTION



## SUMMARY

In summary, a number of noise sources have been described briefly, as well as some of the causes and cures. The best approach to high performance data bus integration is noise prevention and test verification. After the fact cures are expensive and often ineffective when cable routing, coupling, grounding and shielding must be reconfigured throughout the system. Keep in mind that intersymbol interference is a major source of increased error rate and that the most common cause is non-linearity of the trahsmission line. Particular emphasis should be given to reduction in stub lengths insofar as possible. Also, externally induced noise and waveform distortion should be minimized by balanced line-to-ground coupling and low harmonic distortion waveforms.

# MULTIPLEX DATA BUS APPLICATIONS IN AIRCRAFT SIMULATORS

Chris G. Horattas

Goodyear Aerospace Corporation
Defense Systems Division
1210 Massillon Road, Akron, Ohio 44315

## ABSTRACT

This paper describes an adaptation of certain airborne aircraft systems to a ground based aircraft simulator by duplicating or emulating the aircraft multiplex bus. The considerations that led to such adaptation, the problems that arise and the means by which they may be resolved are discussed.

A special data bus controller that makes the use of on-board aircraft equipment in a simulator, without modifications to the hardware or software is described.

A specially designed multiplex data bus for simulator applications is also discussed.

## INTRODUCTION

Current Air Force simulator procurements require that operational flight programs developed specifically for the airborne computer, be capable of being used in the simulator without modifications; furthermore, that changes to such programs necessitated by aircraft updates, be capable of being incorporated into the simulator, affecting its operation in the same way such changes affect aircraft operation.

To meet these requirements, simulator manufacturers resorted to emulators for the on-board flight computers, or to the actual use of the on-board computers in the simulator. Other aircraft systems, particularly the tactical displays, have grown in complexity, so that their simulation can best be achieved by using actual aircraft systems. These aircraft equipment, when used in the simulator, must be interconnected exactly as if they were in the aircraft. Thus, the aircraft multiplex data bus must be transplanted to the ground based simulator. It would seem like a relatively simple matter to purchase the on-board computer



PRECEDING PAGE BLANK

with its operational software, the aircraft equipment and an aircraft multiplex data bus, interconnect them and produce a simulator. It would also be easy to conclude that the simulator manufacturer's data bus problems have been solved by the airframe manufacturer.

Unfortunately, the requirement of using on-board equipment in simulators does not always work in the direction of simplifying the simulation task. Sometimes it solves one problem at the expense of creating another. How, for example, would unmodified airborne equipment by physically interfaced with simulated aircraft equipment? Are signals electrically compatible? How will timing and synchronization be accomplished?

By far the majority of the systems used in the aircraft are simulated in the aircraft simulator. Black boxes and active control panels may be replaced by facsimiles, that only resemble their aircraft counterparts. The aircraft operational software requires that all peripheral devices, normally connected to the data bus, be connected and operating. This requirement must be met, whether such software is installed in an aircraft, or in a simulator. Control words generated by the on-board computer must be accepted and responses generated by all equipment connected to the bus.

Those devices that normally are connected to the aircraft bus, but omitted in the simulator, are designated as "ghost" devices. A special purpose controller designated as Ghost Devices Controller (GDC) is designed to provide the necessary responses to the bus controller in lieu of the "ghost" devices. This controller is discussed in great detail in this paper.

In addition to the transplanted airborne equipment and aircraft bus, there exists in the simulator a number of simulated systems, that consist of modified hardware and special software modules, which together create the effects that the real aircraft systems would produce. A specially designed bus that interconnects these simulated systems, designated as Simulator Data Bus (SDB), is also covered in this paper.

## GENERAL BACKGROUND ON SIMULATION

Since this paper is primarily concerned with multiplexed data buses pertaining to aircraft simulators, it is considered appropriate to describe, in general terms, a typical aircraft simulator.

Figure 1 is a simplified block diagram of a typical aircraft simulator, identifying its major components. An aircraft simulator consists mainly of four major components. They are: 1) the trainee station, 2) the instructor station, 3) the computation system and, 4) the environment simulation complex. Each of the four components are discussed as follows:

1) The trainee station. The trainee station consists primarily of a cockpit, which resembles the cockpit of the simulated aircraft down to its most minute detail. This is the station at which a pilot trainee will gain familiarity

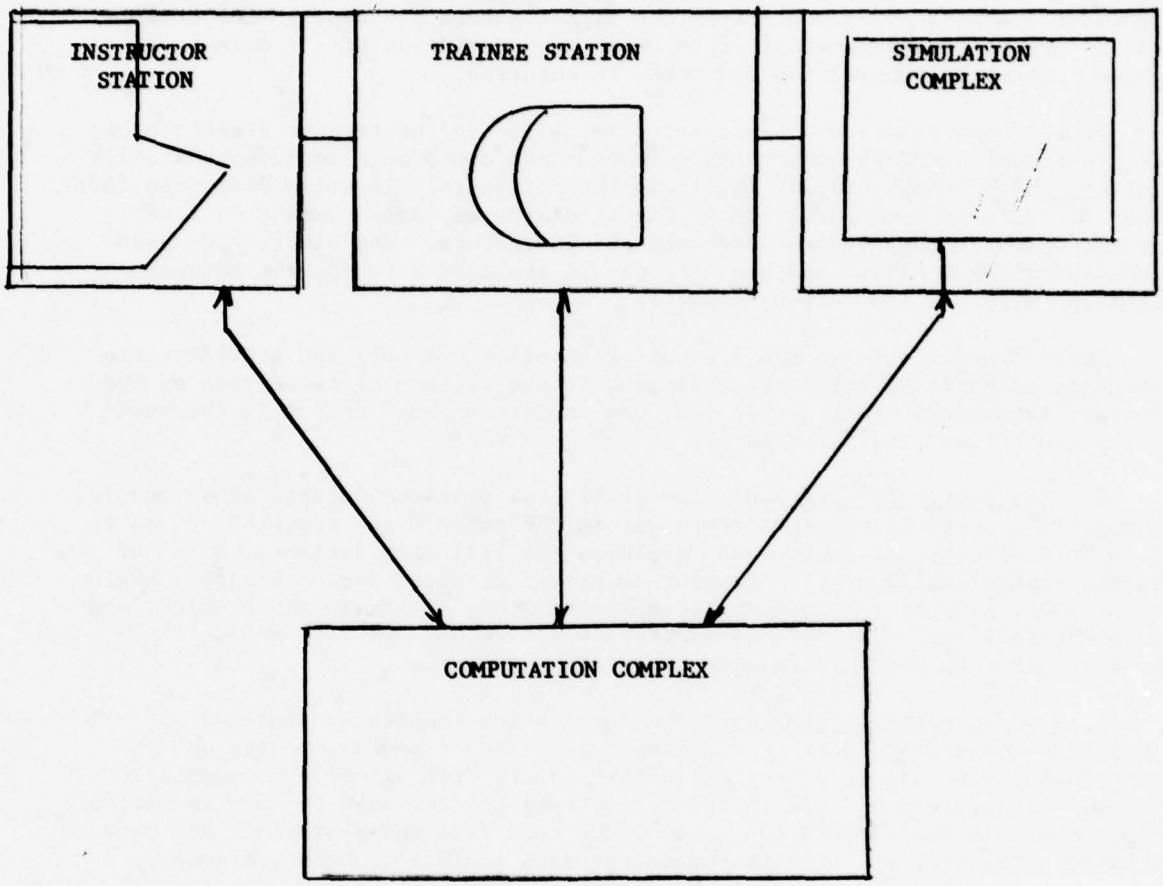



FIGURE 1. BLOCK DIAGRAM OF AN AIRCRAFT SIMULATOR

with the aircraft and acquire skills in flying and operating the real aircraft. It is essential, therefore, that all controls and indicators in the simulator cockpit, including visual and aural displays, function exactly as they would in the actual aircraft. The trainee must derive the same sense and feel from the simulator cockpit, as he would from the actual aircraft cockpit. The trainee station may be mounted on a platform, that can provide up to six degrees of motion - three in translation and three in rotation.

Some of the panels and black boxes installed in the trainee station cockpit are actual aircraft components. Among these components could be the radar display, the Heads-Up Display (HUD) and the Horizontal Situation Indicator (HSI). Other aircraft systems installed in the simulator may only consist of front panels, which contain various controls and indicators. The electronics associated with these panels and black boxes are replaced with program modules operating within a simulation computer.

These program modules receive cues from pilot controls and provide corresponding indications, which are analogous to the effects of these cues on the aircraft aerodynamic characteristics. An example of such device is the navigation control and indicator panel.

2) The instructor station. The instructor station consists of a console, on which all controls and indicators within the cockpit are repeated, so as to allow an instructor to monitor and evaluate the trainee's actions. A set of special controls allow the instructor to preset or alter parameters of a training mission, e.g. to initialize a training mission, to freeze the problem, and to inject failures. An inter-communication system also exists, which allows the instructor to communicate with the trainee by voice.

3) The computation complex. The computation complex consists of all simulation computers and related peripherals. The on-board computer and the Ghost Device Controller, discussed in this paper, are part of the computation system. All trainee and all instructor actions produce cues to the simulation programs within the computation system. Outputs from these programs are used to drive actuators, or activate indicators and displays at both stations.

4) The simulation complex. The simulation complex comprises all hardware required to provide realistic effects, which enhance the simulation. Aural tones, synthetic targets generation, feel forces and cockpit motion, are some of the outputs of the simulation hardware complex.

It should be apparent by now that a simulator, in addition to simulating the aerodynamics of the aircraft, must also simulate the environment. Ground and airborne active emitters, visual scenes, and ground radar returns are but a few of the environmental requirements in the simulator. These combined requirements create an enormous data traffic on the data bus at near real time speeds.

THE GHOST DEVICES CONTROLLER

1. General. The Ghost Devices Controller (GDC) is a specially designed device, which is connected to the aircraft multiplex bus to compensate for all "ghost" devices. "Ghost" devices are those devices that are normally connected to the bus within the aircraft, but are not necessary in the simulator environment. The GDC is programmed to respond to all control words directed by the airborne computer to the "ghost" devices. It will accept or supply data and status words peculiar to the "ghost" devices, while it appears transparent to the on-board computer and its operational flight program. Figure 2 is a block diagram, which compares the airborne multiplex data bus to the same bus in the simulator. The position of the GDC in relation to other devices on the bus is shown in Figure 2B.

The GDC also provides a data link between the on-board computer and the simulation computer complex.

2. GDC Description. The GDC contains driving and receiving capability which is compatible with the aircraft multiplex data bus. This compatibility is prescribed by the specification that governs all devices, which are connected to the aircraft multiplex data bus.

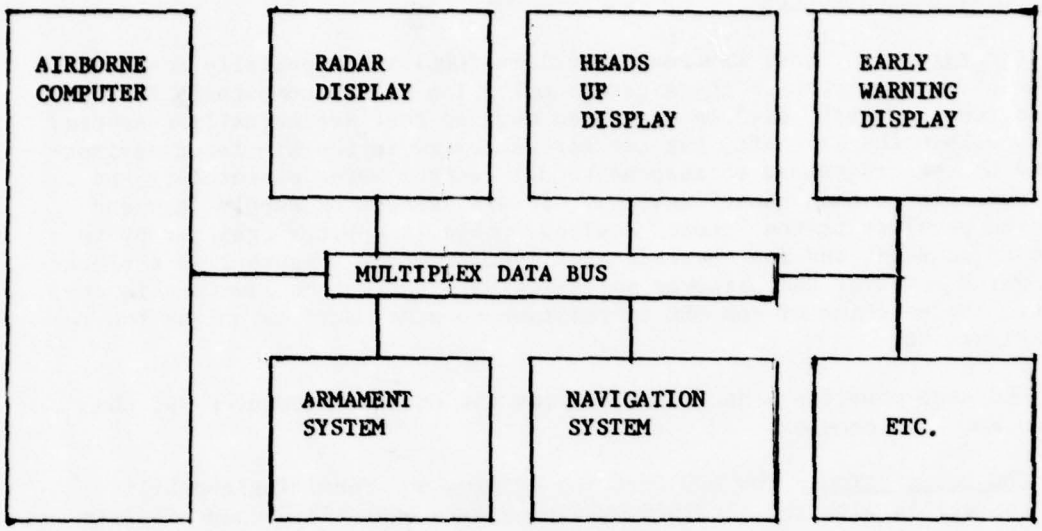
The GDC is programmed to receive all control words emanating from the on-board computer. For control words intended for one of the "ghost" devices, the GDC assumes an active role. It accepts, or furnishes, data and control words depending on the mode of operation. For control words to devices physically connected to the bus, the GDC assumes a passive role. It eavesdrops on the bus, monitoring and storing control, status and data words for subsequent transfer to the simulation computer.

Finally, the GDC has the capability of assuming the role of a bus controller in a back-up mode. This occurs whenever the on-board computer loses control of the multiplex data bus.

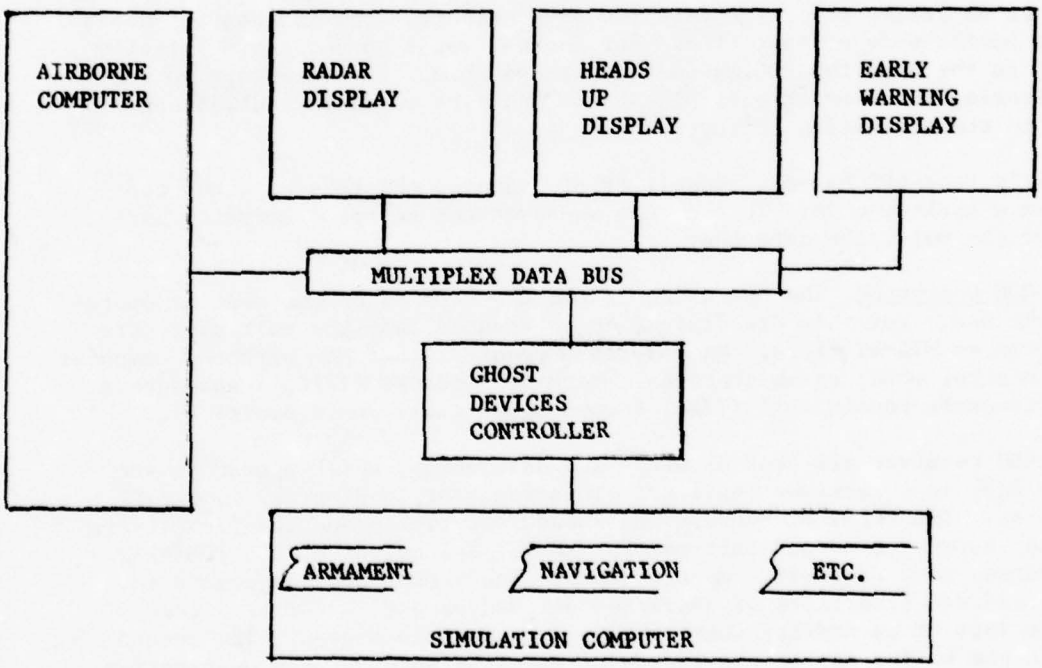
3. GDC Operation. The operation of the GDC depends on the mode of operation of the bus. For this discussion, it is assumed that the multiplex data bus conforms to MIL-STD-1553. MIL-STD-1553 requires that the airborne computer issues a control word, which contains a terminal address field, a subaddress field, a transmit/receive bit (T/R), a word count field and a parity bit.

The GDC receives all control words and data words, which appear on the multiplex bus, in a receiver register. This register is a serial in parallel out register. The terminal address and subaddress fields and the T/R bit are used as an address to a read only memory (ROM). The output of the ROM is a vector address to a read/write memory (RAM). The vector address points to the first address of a block of addresses defined as a data buffer. The buffer consists of an address, where the control word is stored. The second address in the buffer may be the source of a status word, or the destination of the first word of data, depending on the mode of operation.

Several modes of operation prescribed by MIL-STD-1553 are described as follows:




(2A) AIRBORNE SYSTEMS




(2B)

FIGURE 2. AIRCRAFT SYSTEMS IN THE SIMULATOR

a) The airborne computer transmits data to a connected device, b) the airborne computer requests data from a connected device, c) the airborne computer transmits data to a "ghost" device, d) the airborne computer requests data from a "ghost" device, e) the airborne computer commands one connected device to transmit data to another "ghost" device and g) the airborne computer commands intercommunication between a connected device and a "ghost" device. The operation of the GDC for each of these modes varies slightly as described in the following paragraphs and illustrated on Figure 3 for each mode.

a) Whenever the airborne computer commands a connected device to receive data, the GDC stores the control word in the vector address. The connected device responds with a status word, which the GDC stores in the second address relative to the vector address. All data words sent by the airborne computer to the device are received by the GDC and stored in subsequent RAM addresses for a later transfer to the simulation computer.

b) Whenever the airborne computer commands a connected device to supply data, the GDC again stores the control word in the vector address; the transmitted data are received by the GDC and stored in subsequent addresses; the status word generated by the responder device is stored in the last word of the buffer for later transfer to the simulation computer.

c) Whenever the airborne computer commands a "ghost" device to accept data, the vector address will point to the status word of the "ghost" device, which is read out and transmitted first. This status word is prestored in the GDC RAM by the simulation computer. Subsequent data emanating from the airborne computer are received by the GDC and stored in subsequent RAM addresses.

d) Whenever the airborne computer commands a "ghost" device to supply data, the vector address will point to the first of a block of addresses in the GDC RAM, where data is pre-stored by the simulation computer. The last word transmitted, which is the last word in the block, will be the status word. The status word is also preset by the simulation computer into the GDC RAM.

e) Whenever the airborne computer commands one connected device to transmit data to another connected device, the GDC eavesdrops on the bus and stores the computer control word to the receiver device and the receiver device's status word in the first two vector addresses. The computer control word to the transmitter device is stored in the third word, and the data are stored in subsequent words, followed by the status word of the transmitting device.

f) Whenever the airborne computer commands one "ghost" device to transmit data to another, the two control words are received and stored by the GDC, but only status words will be generated by the GDC to the bus. The vector address of the "ghost" device, that will receive data, points to the RAM address where the first control word is stored, followed by the receiver ghost devices status word, which is transmitted first. The second control word will provide the vector to where the second control word is stored, followed by the status word of the transmitting "ghost" device, which is sent to the bus next. No data transfers are required in this mode.

| ADDRESS | CONTENT | REMARKS |
|---|---|---|
| VA | CONTROL WORD | AIRBORNE COMPUTER COMMANDS A CONNECTED DEVICE TO RECEIVE DATA (GDC PASSIVE) |
| VA+1 | DEVICE STATUS WORD | |
| VA+2 | FIRST DATA WORD | |
| ⋮ | ⋮ | |
| VA+WC+2 | LAST DATA WORD | |
| VA | CONTROL WORD | AIRBORNE COMPUTER COMMANDS A CONNECTED DEVICE TO TRANSMIT DATA (GDC PASSIVE) |
| VA+1 | FIRST DATA WORD | |
| ⋮ | ⋮ | |
| VA+WC+1 | LAST DATA WORD | |
| VA+WC+2 | DEVICE STATUS WORD | |
| VA | CONTROL WORD | AIRBORNE COMPUTER COMMANDS A GHOST DEVICE TO RECEIVE DATA (GDC ACTIVE) |
| VA+1 | GHOST DEVICE STATUS WORD | |
| VA+2 | FIRST DATA WORD | |
| ⋮ | ⋮ | |
| VA+WC+2 | LAST DATA WORD | |
| VA | CONTROL WORD | AIRBORNE COMPUTER COMMANDS A GHOST DEVICE TO TRANSMIT DATA (GDC ACTIVE) |
| VA+1 | FIRST DATA WORD | |
| ⋮ | ⋮ | |
| VA+WC+1 | LAST DATA WORD | |
| VA+WC+2 | GHOST DEVICE STATUS WORD | |
| VA | CONTROL WORD TO RECEIVE DEVICE | AIRBORNE COMPUTER COMMANDS ONE CONNECTED DEVICE TO RECEIVE AND ANOTHER TO TRANSMIT (GDC PASSIVE) |
| VA+1 | RECEIVER DEVICE STATUS WORD | |
| VA+2 | CONTROL WORD TO TRANSMITTER DEVICE | |
| VA+3 | FIRST DATA WORD | |
| ⋮ | ⋮ | |
| VA+WC+3 | LAST DATA WORD | |
| VA+WC+4 | TRANSMITTER DEVICE STATUS WORD | |
| VA | CONTROL WORD TO RECEIVER GHOST DEVICE | AIRBORNE COMPUTER COMMANDS ONE GHOST DEVICE TO TRANSMIT DATA TO ANOTHER GHOST DEVICE (GDC ACTIVE) |
| VA+1 | RECEIVER GHOST DEVICE STATUS WORD | |
| VA+2 | CONTROL WORD TO TRANSMITTER GHOST DEVICE | |
| VA+3 | TRANSMITTER GHOST DEVICE STATUS WORD | |
| VA+4 | | |
| ⋮ | | |

NOTE: VA DENOTES VECTOR ADDRESS
WC DENOTES WORD COUNT

FIGURE 3 - GDC RAM MEMORY MAP

g) Whenever the airborne computer commands one connected device to communicate with a "ghost" device, or conversely, the receiving device will supply its status word before the data. The transmitting device will supply its status word after the data. The GDC operates similar to previously described modes, by ordering the control words, status words and data words in memory so as to follow the sequence specified by the bus convention.

Figure 4 is a block diagram illustrating the GDC logic. The receiver register output after a control word is received, will consist of the unit address, subaddress and T/R bit. This output forms the address to the ROM. The output of the ROM is the vector address to the RAM which is preloaded into a memory address register (MAR). As data is received, or transmitted, the MAR is incremented and the word counter is decremented. The end of transmission will occur when the word count reaches zero.

4. Simulation Computer Interface. The simulation computer interface to the GDC is similar to the interface between the ROM and RAM. A parallel address is received from the computer channel, which accesses directly the RAM memory (DMA), to preset data and status words, or to sample incoming data and status words. This interface is interlocked with the multiplex data bus in order to prevent interference with ensuing transmissions.

A second DMA interface from the simulation computer may be provided, which connects to the airborne computer Ground Support Equipment (GSE) port. This port allows loading and verifying functions to the airborne computer. During operation, the simulation computer can gain direct control over the airborne computer to simulate mission freeze, initialize or fail by using the GSE port interface.

5. Timing. In accordance with the MIL-STD-1553, data and status words must be obtained in the three microsecond SYNC time. Even though this time is adequate for accessing the RAM, an additional six microseconds could be provided by employing a buffer register. At shift count 11, the terminal address, subaddress and T/R bit are available in the receiver register and they could be transferred in parallel to the buffer register. While the word count and parity bits are being shifted, the vector address is pipelined to the RAM and the response word is in the Transit register even before the entire control word is received.

To assure that data between the airborne computer and the simulation computer propagate throughout the trainer in the same program cycle, the DMA interface between the simulation computer and the GDC is done twice during the cycle time of the airborne computer.

SIMULATOR MULTIPLEX DATA BUS

To accommodate the enormous data traffic between the computation system and the simulation hardware, a second multiplex data bus has been incorporated in the simulator. This bus differs from the airborne multiplex data bus in that, it can accommodate devices of dissimilar word formats. Unlike the airborne bus which consists of a single data line, the simulator multiplex data

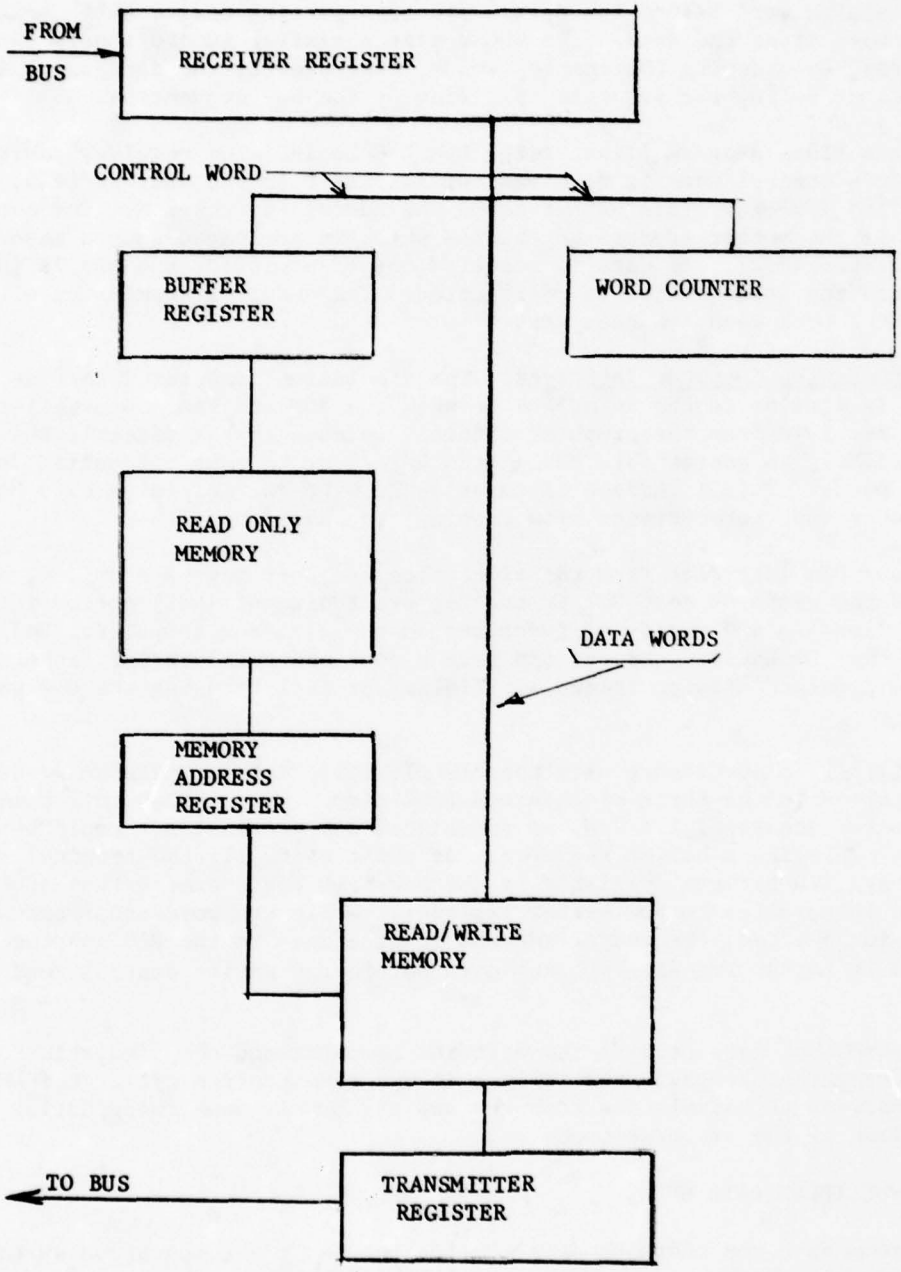



FIGURE 4 - GHOST DEVICES CONTROLLER FUNCTIONAL DIAGRAM

bus consists of the data line, a synchronizing shift clock, a data envelope and a command indicator pulse. The four signals transmission lines consist of twisted pairs driven by tristate bus drivers. Due to the fact that the clock is distributed to the various simulated devices, encoding into a bi-phase code such as the Manchester Code is not necessary. Incorporating encoding and decoding electronics into each device connected to the bus would be unwieldly, especially if the device is a single board.

The simulation multiplex data bus described has two main features: 1) it reduces the multitude of interconnecting cabling by its distributed I/O feature and 2) it reduces system noise and power requirements by its addressing structure.

1) Distributed I/O. The distributed I/O scheme allows a set of data acquisition modules and conversion hardware to be located at each remote station. Where the moving platform is involved, the resulting reduction of cabling, from several thousand conductors to four conductors, is a very desirable aspect. All data are transferred serially by bit, serially by word between the two stations and the computation system on the simulator multiplex data bus.

2) Addressing Structure. The addressing in the simulation multiplex data bus is so designed as to minimize the number of drivers and receivers that can be active at the same time. In fact a single driver and a single receiver can be enabled at one time, resulting in lower power dissipation and lower noise generation in the simulator.

Because of the serial transmission of data, several other features result: The data can be easily formatted into words, bytes or bit groups of any predetermined format, thus reducing packing, unpacking and bit processing by the computer I/O programs, as well as making interfacing of devices of varying bit formats possible. Furthermore, the serial data transmission scheme lends itself to ease of closed loop testing, required by readiness and maintenance tests.

Figure 5 is a block diagram that illustrates the Simulator Multiplex Data Bus. Several levels of addressing are provided. At the computation system, a two-bit address could select one of four stations, with which communication is desired. A single bit can establish the direction of data flow. Another two bits can establish one of four groups of signals (i.e., Analog, Discrete Synchro, etc.). Finally, the remaining bits can decode into a device address, and data word count.

BIT PROCESSING

Command words and data words are transferred in parallel from the channel to an I/O controller. The I/O controller serializes the data onto the multiplex bus. Bit grouping will be accomplished within the I/O controller by the formatting logic. The formatting logic identifies words by a pre-determined sequence and provides variable gates to tailor each transfer to the bit format of the device

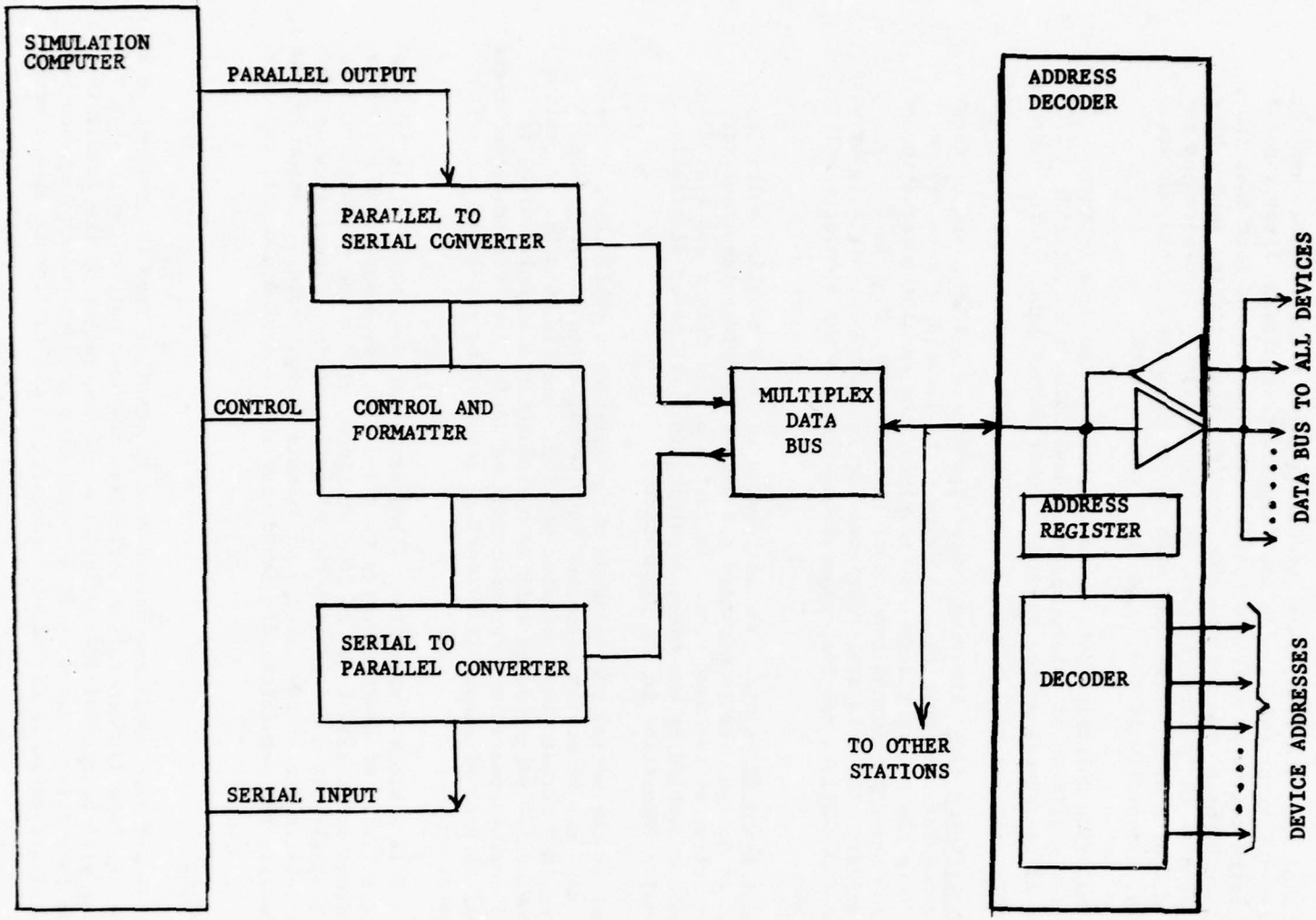



FIGURE 5 - SIMULATION MULTIPLEX DATA BUS BLOCK DIAGRAM

for which data is intended.

Conversely, data originating at the remote devices will be accepted by the I/O controller in serial format and converted to parallel formats of varying bit lengths, before they are transferred in parallel to the simulation computer.

## OPERATION

The multiplex data bus is synchronized to computer handshake signals so as to make external storage of data unnecessary. Commands are identified by a Command indicator handshake signal. The Command words may contain the initial device address, the number of words to be transferred, the direction of data flow and a station sub-address.

Either of two modes of transmission could be selected. 1) The sequential mode, and 2) the random address mode. In the sequential mode one command word precedes a block of several words. Devices are programmed by decoding an incremented address generator located at each station. Only one device can be connected to the multiplex bus at one instant. This mode requires that all devices be updated for each CPU I/O cycle. In the random mode, a command word must be generated for each data transfer. The throughput of the bus appears to be decreased by the random address mode. If however one considers that data from various devices may change less frequently than others, a combination of random addressing and a sequential addressing could be devised that provides optimum operation.

## CONCLUSION

Multiplex data buses in simulators have been successfully utilized with a great number of advantages realized. The most obvious advantage is cabling reduction due to multiplexing, resulting in cost savings, reliability, maintainability, ease of installation and modifications. Serial data multiplexing adapts easily to on-line and off-line tests of the various circuit cards within the system to establish system integrity.

## REFERENCES


1) Goodyear Aerospace Corporation GER 16486, 19 Sept 1977 Smart Interface

2) Goodyear Aerospace Corporation GER 14690, 19 Sept 1977 Data Formatter

3) MIL-STD-1553A, 30 August 1973

# 1553B: HOW DO YOU KNOW YOU'LL PASS THE NOISE TEST?

Robert M. Salter

Sperry Univac

ABSTRACT

1553B requires a figure of merit to be met; operation in 200mv RMS of white noise with word error rate of $10^{-7}$ or better. How does one reliably run this test in a laboratory and how does one analyze a given design for noise rejection? The laboratory test requires special coupling of white noise onto the line and care in connecting monitor equipment. Analysis of the effect of noise requires accounting for the band limit of 4 MHZ and considering the various ways in which noise can cause a problem, e.g., during rise times, peaks, and dead times.

PRECEDING PAGE BLANK

## I. INTRODUCTION

A new error rate test faces the designer of a MIL-STD-1553B interface. The essence of the test for a direct coupled system is:

> 200mV RMS of white noise bandwidth 1KHZ to 4MHZ, shall be imposed upon a 3V peak-to-peak signal and the resultant word error rate shall be less than 1 in $10^7$ words.

1553A does not require this figure of merit but instead uses an electromagnetic interference test (MIL-STD-461 and 462). The direct noise test was suggested in the 1976 Proceedings and was actually required for the B-1, AMUX system but generally this will be new territory for 1553 designers. The good news is that the test is not difficult to pass. The bad news is that it is difficult to prove you passed it. The test is somewhat difficult to set up properly and analysis of the effect of noise is non-trivial. But the lab testing should not be ignored and the proper analysis of noise effects may be rewarding (or required). So this paper will hopefully give you a "leg up" on both testing and analysis of noise. It is not really a "How To" manual but rather a guide to what to be aware of.

## II. WHAT IS WHITE NOISE?

The textbook definition of white noise is a signal with an equal mix of all frequencies, randomly phased. Its spectrum therefore is flat (Figure 1A). In the time domain the noise appears as "hash". As the frequency components randomly add and subtract the signal changes amplitude, occasionally achieving large values. The probability of a given sample exceeding a given level is derived from the Gaussian or Normal probability chart (Figure 1B). The graph is used as follows: the total area under the curve is unity. $\sigma$ corresponds to the RMS value of the white noise or the standard deviation. If $\sigma$ = 200mV the probability of a given sample exceeding 1v ($5\sigma$) is the shaded area of Figure 1b. This value is approximately $3 \times 10^{-7}$ (see Table at end of paper). Three samples in ten million exceed 1v. If the noise bandwidth is much greater than the sampling rate the statistical correlation is zero, so that the probability of two consecutive 1v samples is $(3 \times 10^{-7})^2$. (For high sampling rates this is not true as will be seen later).

## III. PITFALLS OF LABORATORY TESTING OF NOISE

The noise test is not difficult to pass. Each decoder design is immune to certain noise effects and the rest can

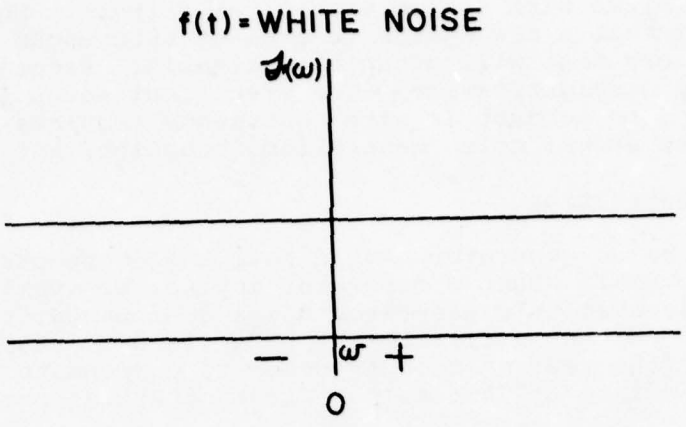



FIGURE 1A) FREQUENCY SPECTRUM OF WHITE NOISE

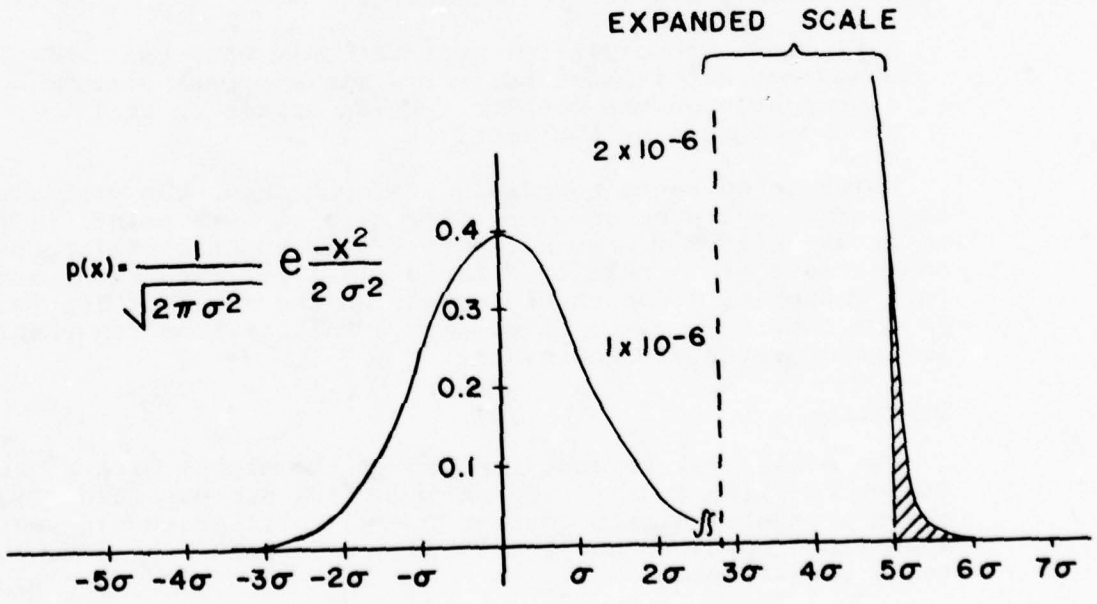



FIGURE 1B) GAUSSIAN NOISE PROBABILITY DENSITY

be minimized with analog or digital filters. However, to confirm that a new design is free of unforeseen flaws, a laboratory test will often be desirable. Programming the test is straightforward. The electrical setup is not trivial and to conduct it with confidence requires attention to three areas: noise generation, coupling, and monitoring.

Noise Generation

The noise generator should have a flat spectrum from 1KHZ to 4MHZ. Such a generator may not be available. (A commonly available generator has a 5MHZ bandwidth, for example). Either a filter must be devised to shape the bandpass or the test must be adjusted to compensate for the "imperfection". This is the first potential problem:

> Pitfall 1: If the noise spectrum is not a flat 1KHZ to 4MHZ, be prepared to prove that the spectrum used is "worse", that it causes more errors than an ideal test.

As an example, if a 5MHZ bandpass is used, 200mV RMS puts only 179mV into the 1KHZ-4MHZ spectrum. 222mV is required to get the proper power in 1KHZ to 4MHZ. Similarly, if your spectrum is not flat you should feel obligated to prove that this defect does not affect test validity.

Once the noise source is chosen a wideband amplifier will likely be required to deliver power to the line. An important consideration is the following:

> Pitfall 2: The amplifier must not only pass the 4MHZ bandwidth but it must deliver a minimum peak voltage, dependent upon the coupler. Noise spikes up to 1.5v should be seen by the receiver.

White noise has a boundless voltage range, but practically the noise generator and amplifier clip at some point. (This point is obvious if the noise is viewed on an oscilloscope). Some errors are a result of large (1v-1.5v) excursions and this amount of noise should appear on the receiver inputs. If the amplifier clips at too low a voltage some remarkably low error rates are achievable.

Coupling

The noise must be superimposed on the signal with a coupler. Direct coupling and transformer coupling are possible. Figure 2 shows a general direct coupler. The amplifier should see its specified impedance and the 1553B output should be attenuated to 3v peak-to-peak.

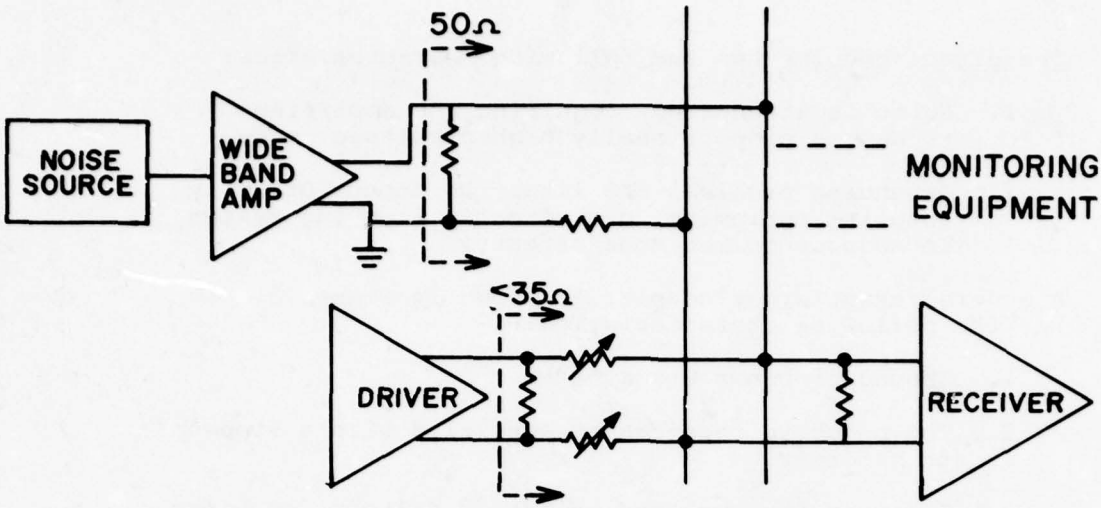



FIGURE 2) DIRECT NOISE COUPLER

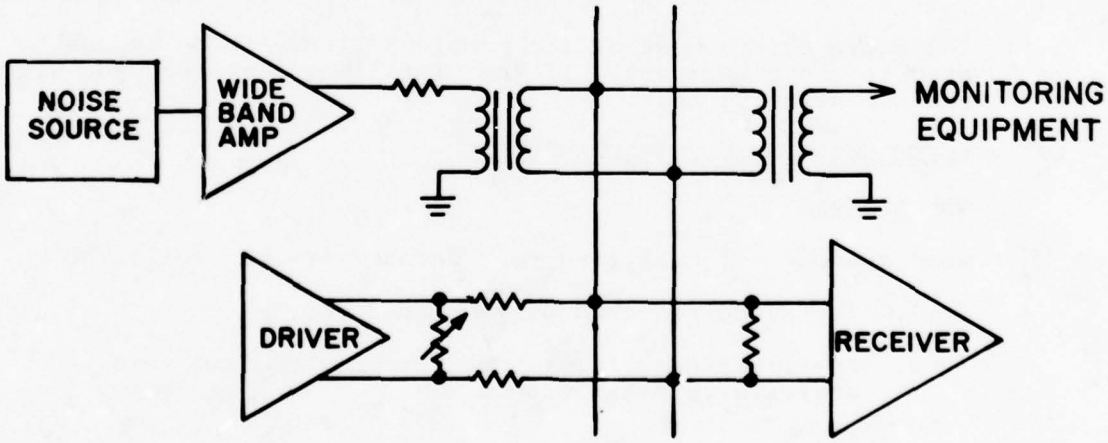



FIGURE 3) TRANSFORMER NOISE COUPLER

The direct coupler has the following characteristics:

1. Noise is attenuated, requiring the amplifier to have a proportionally higher voltage range.

2. Grounding problems are likely to occur. Directly coupling equipment to a normally floating system introduces common mode effects.

A general transformer coupler is shown in Figure 3. It has the following characteristics:

1. Ground problems are avoided.

2. The noise voltage can be amplified with a step-up transformer.

3. The transformers tend to act as filters, perhaps limiting the bandpass.

In view of the above:

Pitfall 3: Don't expect the paper design of a coupler to be sufficient. Prepare rather to break off ground prongs or rewind transformers to achieve a non-distorting coupler.

Monitoring

Implicit in the discussion of coupling is that the line is being monitored for spectrum integrity, clipping, and waveform fidelity. A true RMS meter, spectrum analyzer, and oscilloscope are necessary.

Pitfall 4: Your monitoring equipment will also affect your system. Watch for grounding and coupling problems.

The above covers most of the problems likely to be encountered in the laboratory. If the pitfalls are avoided you will run a convincing test.

## IV. WHITE NOISE THEORETICAL ANALYSIS

Why Analyze?

Some reasons for analyzing your decoder are the following:

1. It may be required by your customer.

2. An "imperfect" laboratory test may require some analysis to back it up.

3. It bolsters confidence in the lab test generally.

4. Noise analysis may influence design.

What the Noise is Really Like

Figure 4 is a photograph of white noise with a 5MHZ bandwidth viewed at 200ns/division. It is not the "hash" normally associated with white noise but is rather slow moving. If we photograph a single trace on the screen once a day and take the value at the center of the screen as our sample, we will find the usual Gaussian distribution of values. (The sample will exceed 1v on the average of three of every ten million samples). Today's sample will bear no relation to yesterday's. If we sample at 10MHZ, twice per division, it is obvious that the samples do relate. Once the noise exceeds a volt it may do so for several samples.

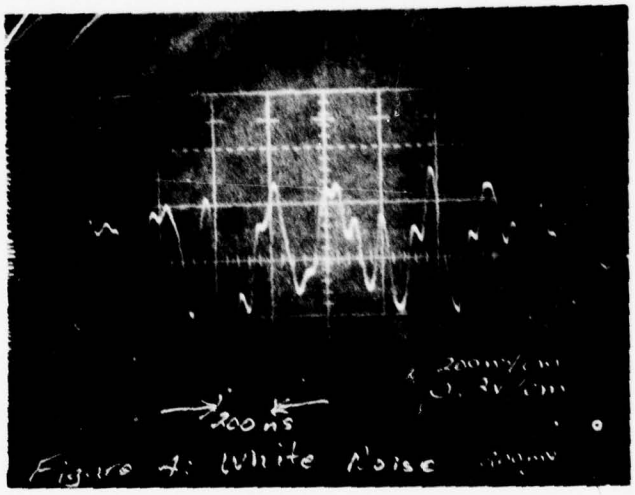


Figure 4: White Noise

Is the probability of a given sample exceeding a volt still a three in ten million? Yes! But the next sample 100ns later has a much higher probability. A more general description of the noise is: the fraction of time white noise dwells above a given voltage is equal to the probability of a given sample exceeding that voltage. E.g. the white noise exceeds a volt 3/10 of the time. As the sampling rate increases more consecutive samples above a volt are caught but there are proportionally more samples below a volt as well.

A second characteristic of the bandlimited noise is that the distribution of energy is uniform throughout the 1K to 4MHZ spectrum. The energy is proportional to the square of the voltage and the noise can be considered, for example, as 4 1MHZ wide spectra at 100mv all superimposed. (Each 1MHZ 100mv band would deliver 10mw to a 1$\Omega$ load for a total of 40mw, which equals $(200mv)^2/1\Omega$).

The Effect of Noise On The Decoder

One answer to this is "No effect, because I have this filter out front.." The filter, however, must pass 500KHZ, 1MHZ, and the rise/fall fairly accurately to get the signal through, and some noise inevitably passes. Assume that the receiver/decoder sees all the noise and consider the effects. Problems that noise could cause include the following:

1. Pulse stretching and shrinking.

2. Spurious zero crossings during peaks.

3. Envelope glitching.

4. Voltage reversal during rise/fall.

For each an approach to answering: "What is the probability of this occurring?" is given.

Pulse Stretching and Shrinking

Consider this pulse stretching problem: the circuit sets an error flag if the first half sync exceeds 2000ns (nominal is 1500ns.) What is the probability that noise will stretch the first half sync of transmission 500ns?

Approach: The noise must be 500ns long and at least 0.5v high (to trip the receiver). The noise in the 1MHZ to 4MHZ band is a mix of sine waves with half periods less than 500ns. This will never produce a pulse 500ns wide (and in fact will make wide pulses less likely). The pulse will be caused by the 1KHZ to 1MHZ band, which has 100mv of RMS voltage. This band produces a pulse 500mv high ($5\sigma$) with probability $3x10^{-7}$, which is alarmingly high.

We can pause here and note that $3x10^{-7}$ represents an upper bound on the probability since there are other frequencies present. But the designer, from this quick estimate, is forced to investigate further the susceptibility of his circuit to this problem (or possibly he has a filter which knocks out low frequencies, in which case he knows he is safe).

Spurious Zero Crossing During Peak Voltage

Assume the circuit fails if a noise spike drives the signal through zero in the middle of the 1.5vpeak. What is the probability of this happening?

Approach: The noise spike must exceed -1.5v which is $7.5\sigma$. This has probability much, much less than $10^{-10}$. Even with a decoder sample rate of 10MHZ (200 samples per word) the chance of a word having this error is much less than $10^{-7}$. Rest easy.

Envelope Glitching

Assume the circuit has an envelope detector which rises if the signal exceeds threshold and drops if the signal is between thresholds for longer than 200ns. What is the probability that noise will cause a "glitch" in the envelope?

Approach: A noise spike 1v high and 200ns wide could possibly force the envelope detector to drop. Only frequencies in the range 1KHZ to 2.5MHZ could ever produce a pulse this wide. This band has 158mv of noise (62% of energy). A sample in this range exceeds 1v ($6.3\sigma$) with probability 1 in $10^{10}$. Is this negligible? No. Assume each pulse of noise is 200ns wide. The noise typically spends 200ns above a volt and $10^{10}$x 200ns (2000 seconds) below a volt. How many 20us words are transmitted in 2000 seconds? Only $10^{8}$ which means one word in $10^{8}$ could have an envelope glitch.

While this is an upper bound and seems safe the calculation indicates that although a given event has low probability the word error rate depends on how susceptible the circuit is to the error. This pulse occurring any time in the transmission could cause an error (unlike the pulse stretching example).

Voltage Reversal During Rise/Fall

Assume the circuit fails if a voltage reversal occurs during a rise/fall (Figure 5). What is the probability of this occurring?

Approach: Finding a meaningful upper bound here is very difficult and the example perhaps reveals the limits of analysis. The simplest model assumes all the power is at 4MHZ. A pulse of only 0.6v causes a reversal if it occurs during a rise/fall (15mv/ns slew rate is required).

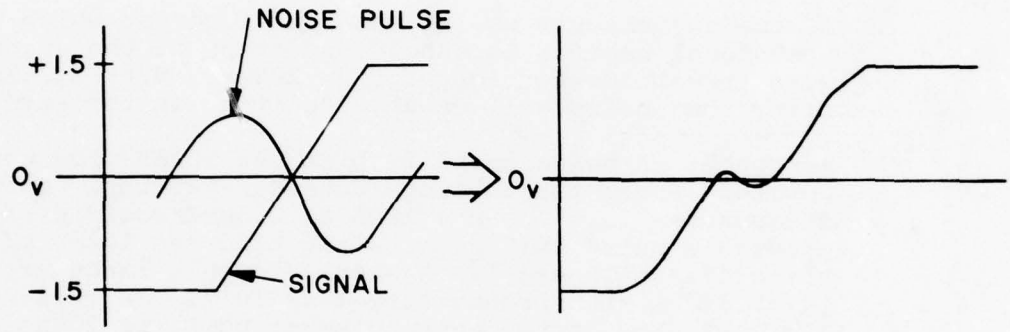



FIGURE 5) SUPERPOSITION OF NOISE AND SIGNAL DURING RISE

Such a pulse (125ns wide) occurs typically once in 31us and even considering the signal is rising or falling about 30% of the time a voltage reversal occurs in roughly one of five words. The models are clearly too crude. The lower frequencies affect probability of a given slew rate, but to calculate the result of all the combinations of frequencies is a task similar to deriving the Gaussian distribution itself.

It is better here to ask a new question: "Do voltage reversals cause any problem?" If you are sampling every 50ns a voltage reversal could cause a glitch, (the question of probability is much more manageable in this case), but sampling every 100ns or 200ns there is no problem at all.

The last example notwithstanding, it is possible to establish meaningful upper bounds on the probability of some important errors and these estimates may directly influence filter or logic design. There are other effects not covered here, such as "Does the circuit hang up if a spurious half sync appears on a dead line? How probable is this?", or "Will noise keep the envelope detect always active? Will this cause errors?" Each circuit has its own susceptibilities and by asking the appropriate question, and finding how much of what frequency noise is relevant, these susceptibilities can be understood.

## APPENDIX

Short Table of Probabilities

| n | $P(V > n\sigma)$ |
|---|---|
| 0 | 0.5 |
| 1 | 0.159 |
| 2 | 0.023 |
| 3 | 0.0014 |
| 4 | $3.2 \times 10^{-5}$ |
| 5 | $2.9 \times 10^{-7}$ |
| 6.18 | $3.2 \times 10^{-10}$ |

Derived from Reference Data For Radio Engineers, 5th Edition, p. 45-19.

MULTIPLEX DATA BUS CONFERENCE

10-12 OCTOBER 1978

DAYTON, OHIO

SPECIAL SESSION

# APPLICATIONS OF FIBRE OPTICS TO MULTIPLEX SYSTEMS

MODERATOR: Francis Lueking
Air-360
Naval Air Systems Command
Washington, DC

# AN OPTICAL-FIBRE MULTI-TERMINAL DATA SYSTEM FOR AIRCRAFT

J.G. Farrington

Standard Telecommunication Laboratories Limited
London Road, Harlow, Essex, CM17 9NA, Essex

A demonstration optical fibre multi-terminal data bus system has been constructed. The system has no central controller, and can use any number of terminals up to the designed maximum. The system has been arranged to operate with a wide range of optical harness layouts.

## 1. INTRODUCTION

This paper will describe some of the results of a development program carried out at Standard Telecommunication Laboratories and funded by MOD (UK).

It is becoming widely accepted that optical fibre data transmission will find application in avionics because of its electrical isolation, wide bandwidth and immunity to cross-talk and radio frequency interference. The principle has already been proved and advantages demonstrated for avionics applications in point-to-point links, much of this preliminary work having been carried out with fibre bundles[1,2]. In the longer term, there also appear to be applications for optical fibre data bus systems. For example, telemetry, display systems or stores management could be suitable choices for first applications. The key optical components - tee-pieces, star couplers, etc - are at an early stage of development, so a system cannot be fully optimized in terms of these components. It is thus too early to attempt to optimize a multi-terminal system for any particular application; it is nevertheless important to start experimental work on multi-terminal systems at this stage. This will ensure that the possibilities of the new technology are included in the long-term data management plans. Otherwise, a framework set up on the basis



PRECEDING PAGE BLANK

of conventional technology, with fibre systems designed to fit in at a later stage, would fail to make full use of the potential.

The other reason for undertaking immediate experimental work is to help direct the development of key optical components such as tee-pieces and star couplers. In addition to using fibre systems to meet conventional requirements more efficiently, there must be new opportunities which would not be considered for electrical systems. Direct optical sensing, without electrical interfaces, is a possible example.

The system approach adopted, in which any number of terminals up to a given maximum (ten in our example) with no requirement for a master terminal, and with great flexibility for the system designer to arrange redundancy, self-checking etc, is likely to be widely applicable. (It can even be used for point-to-point tests, simply by connecting any two terminals). No clear preferred optical highway configuration emerges, and so initial experimental systems should allow this to be flexible.

The system described has been designed to be compatible not only with a 'star' configuration optical harness, which is 'well behaved' from the point of view of the terminals, but with almost any complex highway configuration which may have advantages of considerable redundancy and be adaptable to constraints on layout, maintenance, etc. Such complex highways may be prone to optical echo pulses and other problems, as outlined in Section 2.

It is important to emphasise that the terminal design outlined here is considerably more complex than is envisaged in a typical system. The reason is that this is an experimental system, not an optimized communication link for a particular application, and many of the facilities can be implemented or switched-out at will, in order to gain experience. It is also expected that significant simplifications could be made, although at the expense of demanding improved optical harness performance.

## 2. SYSTEM OUTLINE

### 2.1 General Multi-Terminal Design

The choice of multiplexing technique depends on the capabilities and limitations of the transmission medium, and on the use that is to be made of the entire system. For an

optical fibre several alternatives exist and several solutions are listed below:

i) Time division multiplexing, TDM, where each transmitter in turn has sole use of the optical highway.

ii) Frequency division multiplexing, FDM, where each transmitter uses an optical source, amplitude modulated at a different frequency. This carrier can be amplitude or frequency modulated with information. The total received signal is filtered electrically.

iii) Colour multiplexihg - each transmitter uses a different wavelength source, and filters at the photodiodes separate the signals.

It was decided that the TDM approach was most suitable for systems requiring data interchange between any pair of terminals and so that system will now be discussed in more detail.

## 2.2 The TDM System

### 2.2.1 Synchronisation

For a TDM system, where only one transmitter is allowed to send information at any one time, the timing arrangements are of crucial importance. The most straightforward timing arrangement is through the use of a master terminal, where a single master unit has responsibility for absolute timing, and sends whatever information is needed to the other terminals to maintain the synchronisation of the system. In the extreme, the master terminal can make all decisions about timing and can send a short message to each terminal in turn to start its transmission. This approach can give a very flexible system, where time can be allocated by the master terminal in a way that varies with the amount of information needing transfer, and this leads to very efficient use of the data bus. Unfortunately, this arrangement gives a single vulnerable point in the master terminal where damage or a fault can disrupt the whole system, a situation which can be improved by having several terminals ready to take over from the master if it should fail. This in turn requires a system of priorities being allocated, to specify which terminal will take over first, and involves many problems in deciding whether the first unit has failed or not. To avoid this eventuality, it may be preferable that there should be no master terminal, in which case each terminal must do its own timing and some means must be found of keeping all the terminals in synchronism.

FIGURE 1B) GAUSSIAN NOISE PROBABILITY



This approach was adopted and will be described in more detail in Section 2.3.

One further consequence of this 'autonomy' of the terminals is that the data transmission capacity of them must be predetermined, and cannot easily be changed to meet different network loadings. This approach is thought acceptable for a fibre-optic network as the high bandwidth available means that excess data capacity can be made available at low cost.

2.2.2 Coding

The choice of coding in an optical system can considerably effect the rest of the design. For example, the choice of a PPM code would enable a low-duty-cycle optical source to be used, or a code with no low frequency content would allow an a.c. coupled receiver to be used. Because of the rapid changes of signal level expected at a receiver, it is desirable that the chosen code should allow fast a.g.c. to be applied. Another requirement is that the data clock can be easily extracted from the data, preferably without the use of a phase-locked loop. This is necessary because each received message will be at a different phase due to the different path lengths involved. A short preamble can be included on each message to allow the receiver decoder to settle down, but if this is too long then the system becomes very inefficient.

The most appropriate code would seem to be bi-phase (Manchester) (see Fig.2.1) as this offers several advantages. There is a data transition at the centre of every clock interval which makes clock extraction easy, it only being necessary to time from one cycle to the next with a delay element such as a monostable. The bi-phase code is also arranged to have very little low-frequency content, whatever the data being sent. This occurs because the code has a constant average value for ones and noughts. This allows fast a.g.c. and a.c.-coupled receivers to be used, as stated above.

This code is most ideally suited to bipolar applications where differential signalling is employed. Unfortunately, this cannot easily be achieved with a single optical fibre and so a unipolar version must be used, introducing the disadvantage of needing a threshold shift if the signal strength changes, as illustrated in Fig. 2.1.

## 2.3 Proposed System Outline

### 2.3.1 General Description

The system that has been built at Standard Telecommunication Laboratories has been described elsewhere[3] and so only an outline is included here. A block diagram of one terminal is shown in Fig. 2.3.

A time division multiplex (TDM) approach has been chosen where each transmitter in turn has complete use of the optical highway, for a burst period. See Fig. 2.2. A fixed length message is transmitted in that period, complete with an address code to specify the terminal that is to receive it. The transmitters send in strict rotation, and if a terminal is inoperative, or if no information is available for sending, then that period is either left empty or a dummy message is sent. Each terminal contains its own clock oscillator and divider chain which provides all the timing information within the terminal. These clocks must be kept approximately in step so that the transmission periods do not overlap. This has been achieved by making each transmitted message contain the address of the sending terminal, which is received by all other units and used to correct small timing errors that have developed. Special attention has been paid to the problems of synchronisation from switch-on and to the effects of a single faulty terminal.

### 2.3.2 Basic Design Parameters

The following parameters have been chosen for these functional models. They were chosen as examples only, and it is expected that they would be altered in any design to meet a specific application.

| | | |
|---|---|---|
| Number of terminals | - | up to 10 |
| Bit rate of transmission | - | 1.5 Mbit/s |
| Transmission data rate per terminal | - | 100 kbit/s |
| Optical transmitter | - | Bell Northern LED |
| Optical receiver | - | PIN diode (Hewlett Packard) |
| Optical harness layout | - | to be flexible, with losses up to 30 dB |

The system was designed to have no master terminal.

The message format has been chosen to give a high degree of protection to the service data. The basic arrangement is shown in Fig. 2.2. In the 17 bits allocated to each address, the terminal number is sent twice, and this is followed by an invalid bit sequence to differentiate the address from the data. This degree of protection is necessary as the terminal receivers are active, looking for address codes, at all times, and in between messages are trying to decode random noise.

The data is sent as a block of 128 bits with 8 added parity bits.

### 2.3.3 Fault Protection

The multi-terminal system as described is very flexible, and various self-checking features could be built into the terminals. The timing section ensures that a terminal with a faulty clock divider will shut down its own transmitter as it will continually find large errors between its own clock and the other terminals. This is probably the most serious type of fault that could occur as it might result in transmission at incorrect intervals, so corrupting other terminals' data.

Another form of protection is the maximum transmission time limit. The period for which the transmitter is active should never exceed a certain limit. This is timed independently, and if the limit is exceeded the transmitter is shut down.

Parity checking is incorporated into the system by inserting one parity bit after every 16 bits of data. This is checked by the receiver so that a message with a parity fault can be rejected.

Message confirmation could be incorporated to provide protection against random errors. This has not been done, however, in the functional models. Confirmation could take the form of an added service data section in which a terminal can acknowledge receipt of messages in the last frame. If receipt was not acknowledged the message could be retransmitted by the sending terminal. A suitable code for this is suggested as a ten-bit word (for ten terminals in the system). All bits would normally be zero with one bit in the sequence to represent each terminal. This bit would be sent to indicate that a parity-correct message has been received from that terminal in the last frame. Alternatively, this type of message checking can be built into the external equipment as it may only be required over limited paths.

A more general checking system could consist of a special-purpose terminal which monitors all information on the bus. By checking for parity correct messages, and correct transmitter sequence, etc, it should be possible for the monitor to isolate faulty terminals. The monitor could then be given the power to shut down faulty units. If this was implemented care would have to be taken that the overall reliability was not reduced.

## 3. COMPUTER SIMULATION

### 3.1 Objectives

The computer simulation was carried out at an early stage in the project to investigate the synchronisation behaviour of the proposed system. It was considered necessary to verify the expected behaviour in this way as the synchronisation behaviour of a multi-terminal system with no master is very complicated. There are many interdependent feedback loops involved and it was useful to check that no undesirable stable situations have been overlooked. This is particularly important for a multi-terminal system as final testing can only be carried out with confidence when several (at least three) complete terminals have been built.

To serve the purpose the computer model used was the simplest possible representation of the system and not a detailed hardware simulation.

### 3.2 Results of Computer Simulation

The computer simulation is arranged to run from specified start conditions until the synchronisation conditions are satisfied, when it stops and prints the number of frames that have elapsed. Intermediate results can also be printed, and have been used to verify the operation of the program. All the system variables can be preset (timing counter, error counter, enable-disable flag, clock frequency differences) giving an enormous number of combinations, and so only a few can be tried.

To investigate a wide range of possible start-up conditions the program has been arranged to take random values for its timing counter as initial values, and the result of many runs are plotted as a histogram. The start conditions of the error counters have been found to have very little effect on synchronisation times, and so their start conditions are taken as zero.

Fig. 3.1 shows a typical plot produced in this program. It shows a mean synchronisation time of 2.6 frames with a standard deviation of 0.57.

The very short synchronisation times indicated agree approximately with the experimental results, except that the experimental results show a longer tail, with some very long synchronisation times. This was thought to be due to the optical harness used, which was a 'star' configuration. The program assumes that a ring is used, which would give a single usable signal at every receiver, however many transmitters are active. This is because of the sequential losses around the ring. With a star configuration all optical losses are the same and two transmitters will give approximately equal signals at any receiver. This means that when transmissions overlap the result is that no receiver gets a usable signal.

The computer program was altered to try this arrangement, an occasional long synchronisation times resulted.

## 4. EXPERIMENTAL RESULTS

### 4.1 General Operation

The general operation of the four terminals has been checked with the aid of a four-channel oscilloscope. Fig. 4.1 shows the four transmission-time waveforms with the terminals set as 0, 2, 4 and 6 and Fig. 4.2 shows the corresponding signal received by one of the optical receivers. Figs. 4.3 and 4.4 are the same waveforms, but this time taken with the terminals set as numbers 0, 1, 2 and 3.

The action of the timing correction circuits has been closely checked for a two-terminal set. With only two terminals connected, and their clocks adjusted to be very similar, it is possible to observe the small timing corrections taking place.

Another check that has been carried out on all terminals is to connect them as relays by linking digital output to input. If the terminal is then used to relay a message, originated by a word generator at another terminal, the operation of the input and output circuitry is checked. Fig. 4.5 shows the arrangement used.

### 4.2 Error Rate Measurements

An error measuring counter has been built for use with this equipment. It is based on a microprocessor, which controls

a pseudo-random word generator and checking circuit. The microprocessor totalises the errors counted on each message, and also counts the number of complete messages that are 'lost', corresponding to an error occurring in the message address. Using this equipment, it has been verified that under normal operating conditions error rates are extremely low. Under conditions of reduced signal, errors do occur, and Fig. 4.6 shows a plot of error rate against received power for one receiving terminal.

The flattening-off of the bit error rate at low received powers is thought to be due to the fact that lost messages are not included in this count, and so for very low received powers the bit error rate would fall to zero, as the message error rate rises to unity.

The curve for error rate can be extrapolated giving an error rate of 1 in $10^9$ for a received power of -42.5 dBm.

These error measurements have also been used to show the independence of the data paths. With one transmitter's power reduced to introduce errors, the effects of having full strength transmitters operating immediately before and after the test unit have been investigated. Fig. 4.7 shows the signal at an optical receiver, both with and without the full strength transmitters. The reduced strength signal shows as a non-limited output from the optical receiver. No dependence between transmitters has been found, as the monitored error rates were not found to change when the other units were switched on.

## 4.4 Optical Power Budget

The transmitted power from each of the four LEDs used was measured to be greater than 200 μW mean during the transmission burst. This is the optical power launched into a 150 μm core fibre, with NA ∿0.2. Taking the minimum received power necessary to be -42.5 dBm (from Section 4.2 above) this means that the maximum allowable optical loss between any transmitter and receiver is 35.5 dB.

Allowing for ageing and temperature effects in both transmitter and receiver, and allowing for some system margin, optical losses of up to 30 dB would be acceptable.

The dynamic range of the receiver has been shown to greater than 40 dB and so there are no constraints on the optical harness loss differences, so long as the maximum loss is not exceeded over any path.

## 5. DEMONSTRATION OPTICAL HARNESS

A harness has been constructed which can be used to test the system's immunity to echo pulses arising from multiple path and reflection effects.

The harness consists of single fibre 4-way equal-division transmission stars and a set of fibres of various lengths for connection. These components allow the interconnection of the terminals with various degrees of redundancy. See Fig. 5.1.

Typical insertion losses for a 4-way transmission star are <10 dB between any pair of ports. Optical losses between the mixers are only due to connectors as very low loss fibres were used, and their attenuation can be neglected for the length likely to be used within an aircraft.

## 6. CONCLUSIONS

The four-terminal system with no central controller has been shown to work with a range of different optical harness configurations and losses.

The design approach described would seem to offer a viable architecture for a multi-terminal data system, especially for applications where a central bus controller is undesirable.

## 8. REFERENCES

1. Belcher, G., Marshall, D. "Use of Fibre-Optics - Digital Automatic Flight Control Systems". IEEE Trans. Aerospace and Electronic Systems. Vol. AES-11, No. 5, pp 841-850, Sept. 1975

2. Biard, J.R. "Opto-electronic Aspects of Avionic Systems II". AFAL Report TR-75-4S.

3. Farrington, J.G., and Chown, M. "Optical Fibre Multi-Terminal Data Systems for Aircraft". AGARD, Conf. on Optical Fibres, Integrated Optics and their Military Applications, May 1977

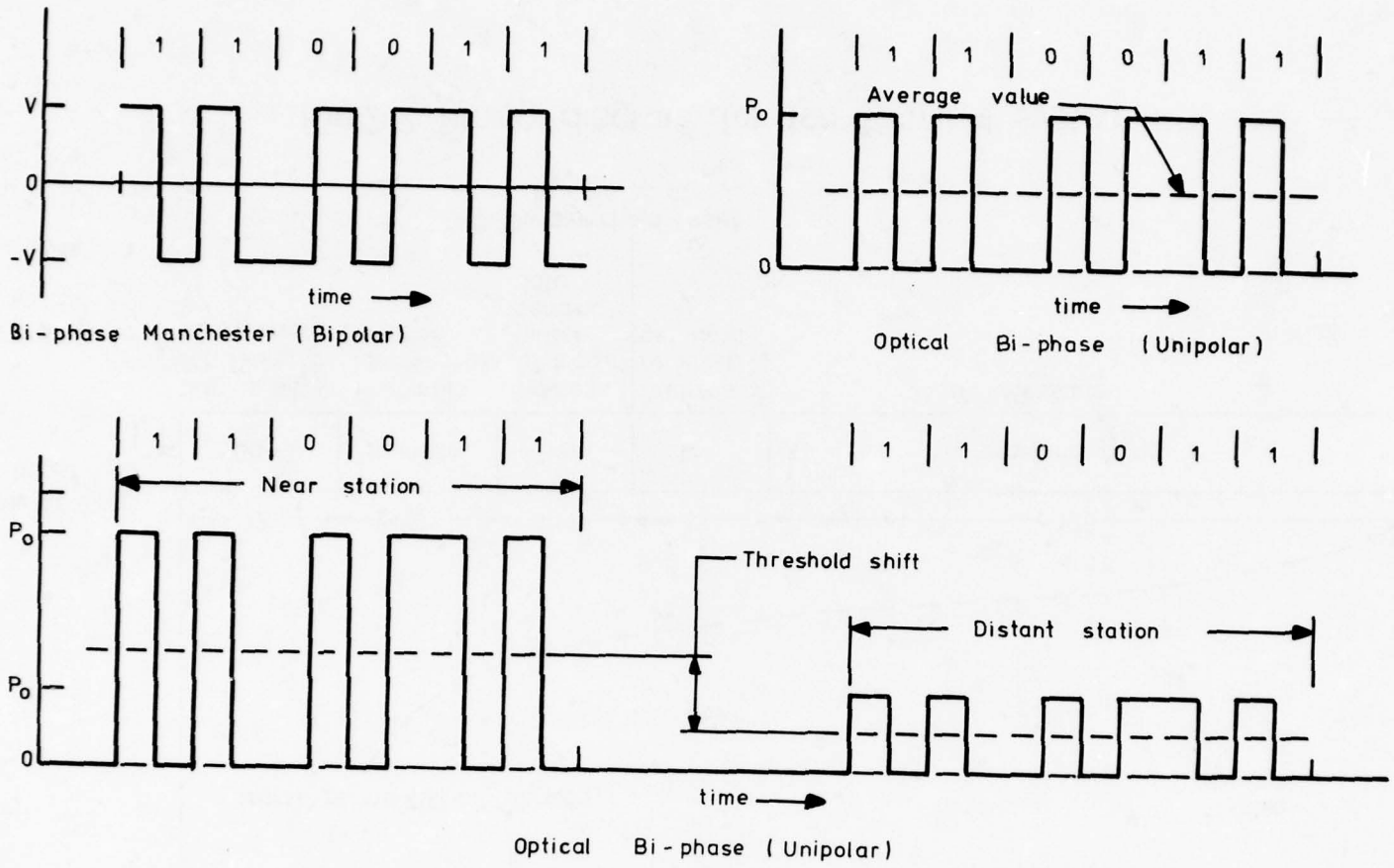



Fig. 2.1 Bipolar and Unipolar bi-phase (after J R Biard ref. 2)

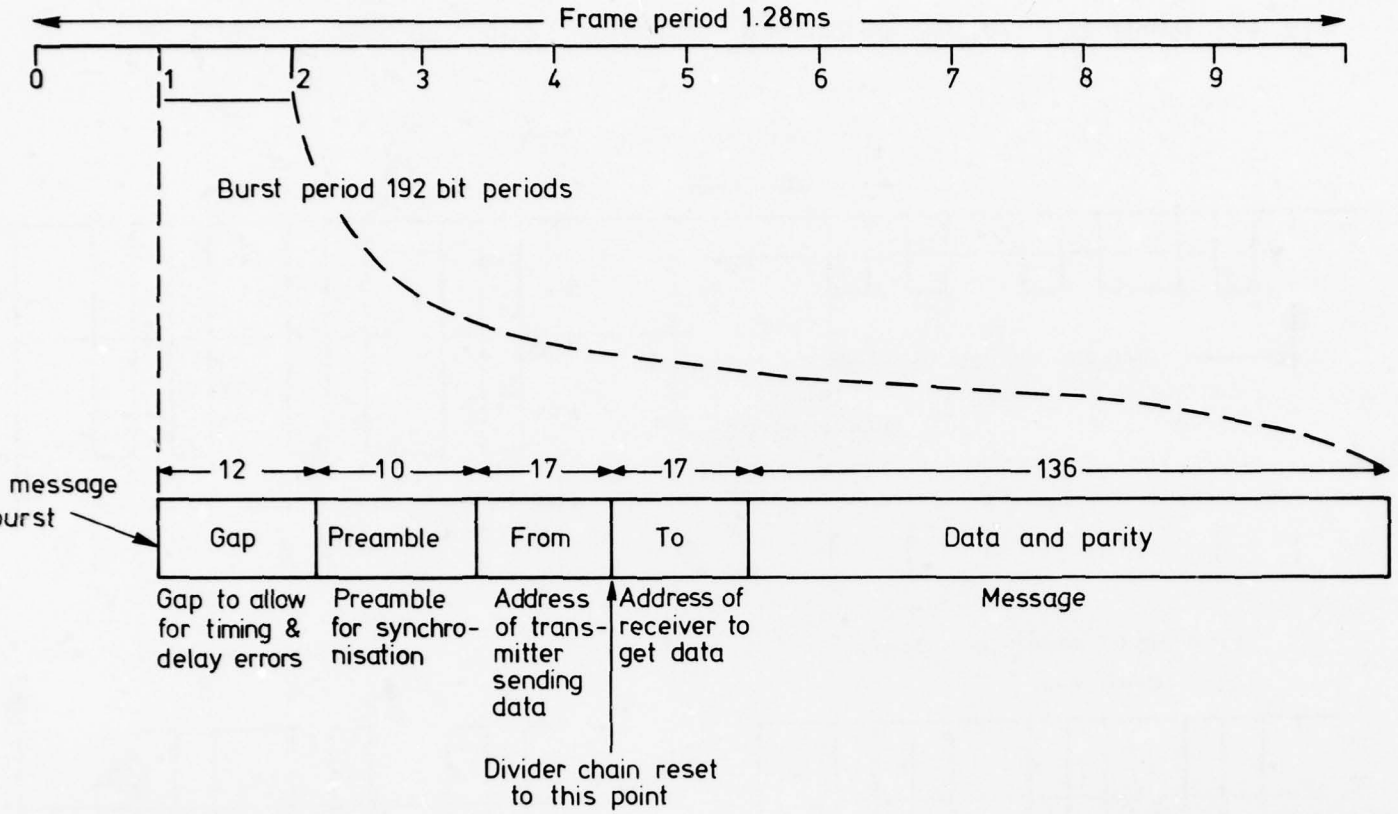



Fig.2.2 Timing diagram for ten terminal system

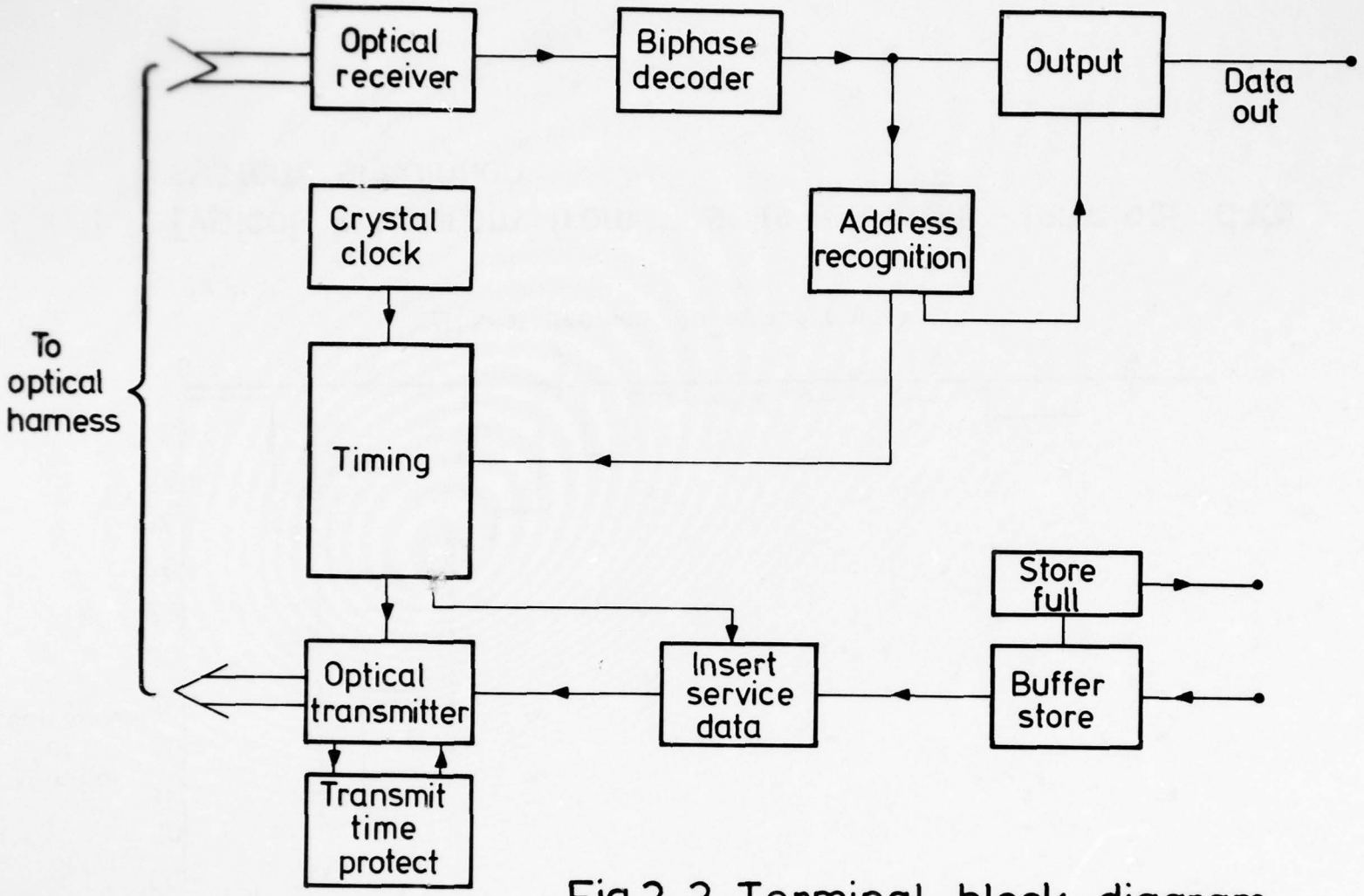



Fig.2.3 Terminal block diagram

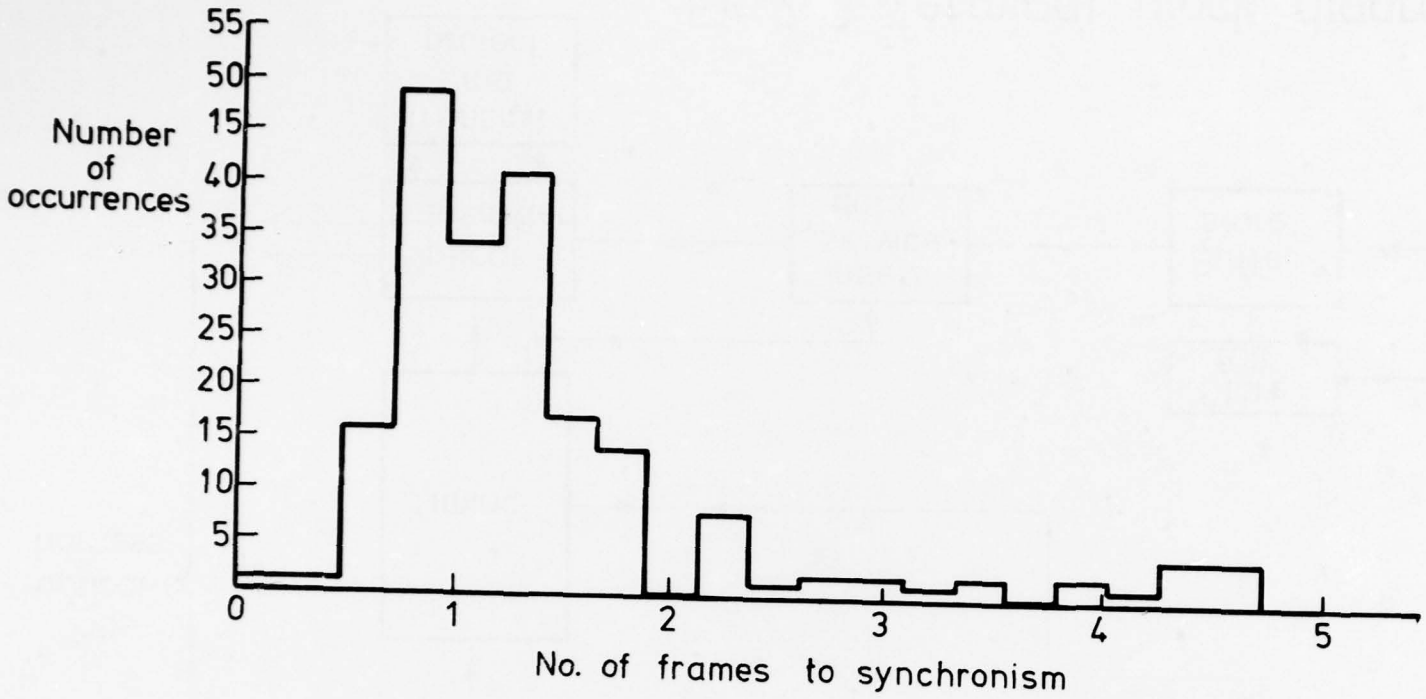



Fig.3.1 Typical histogram from results of multi-terminal data system simulation

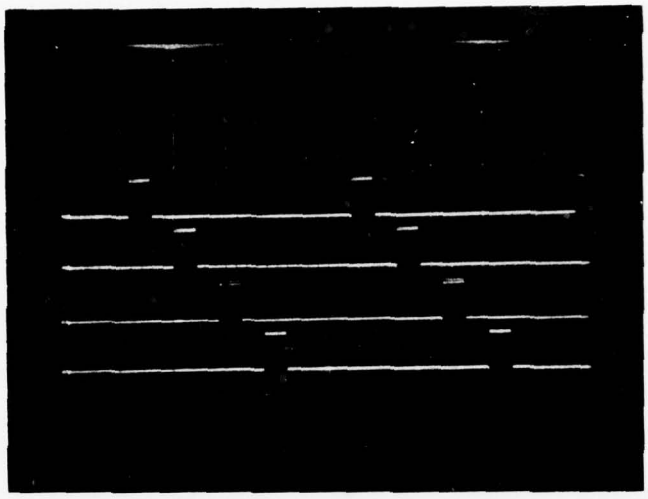


Fig. 4.1 Transmission time intervals from terminals 0, 2, 4 and 6

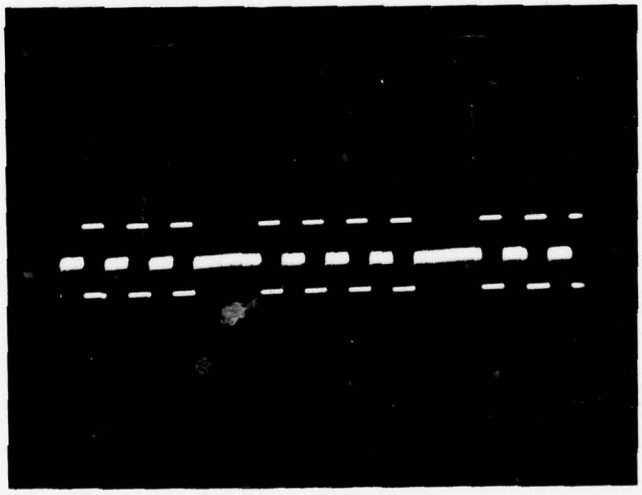


Fig. 4.2 Received signal at one optical receiver for conditions as in Fig 4.1

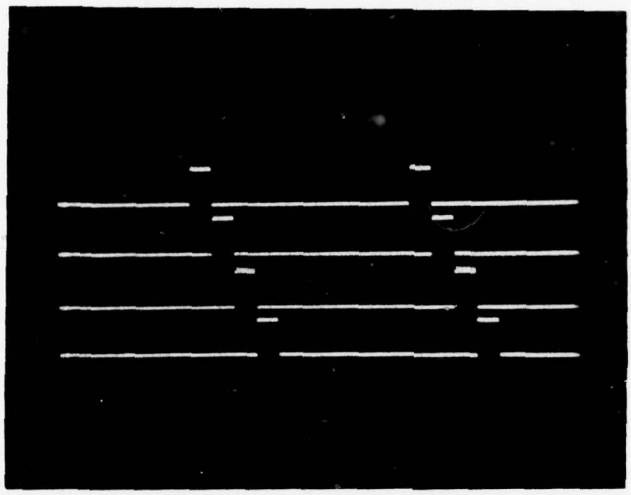


Fig. 4.3 Transmission time intervals for terminals 0, 1, 2 and 3

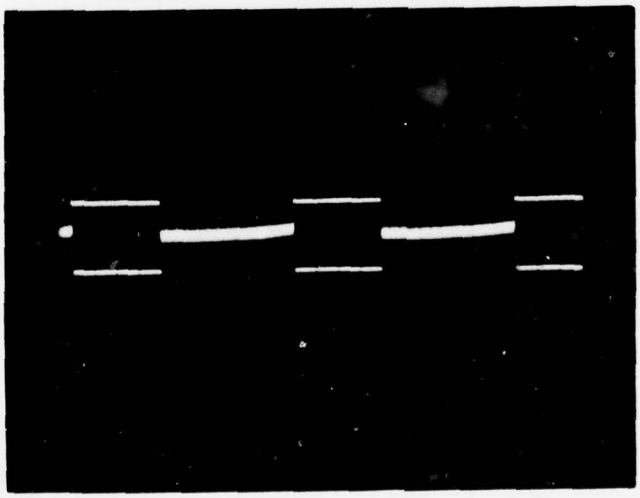


Fig. 4.4 Received signal from one optical receiver for conditions as in Fig. 4.3

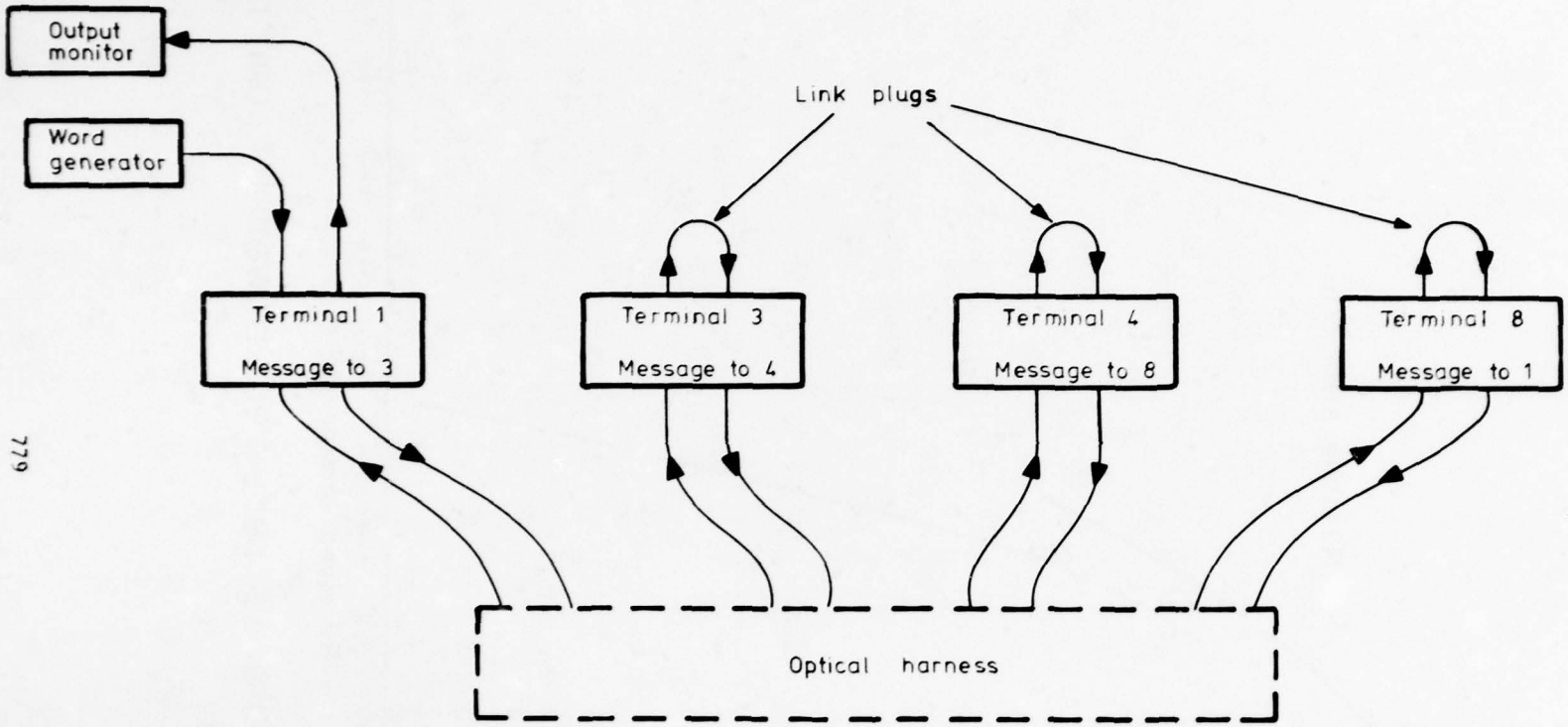



Fig. 4.5 Terminals connected as relays for checking input and output circuits

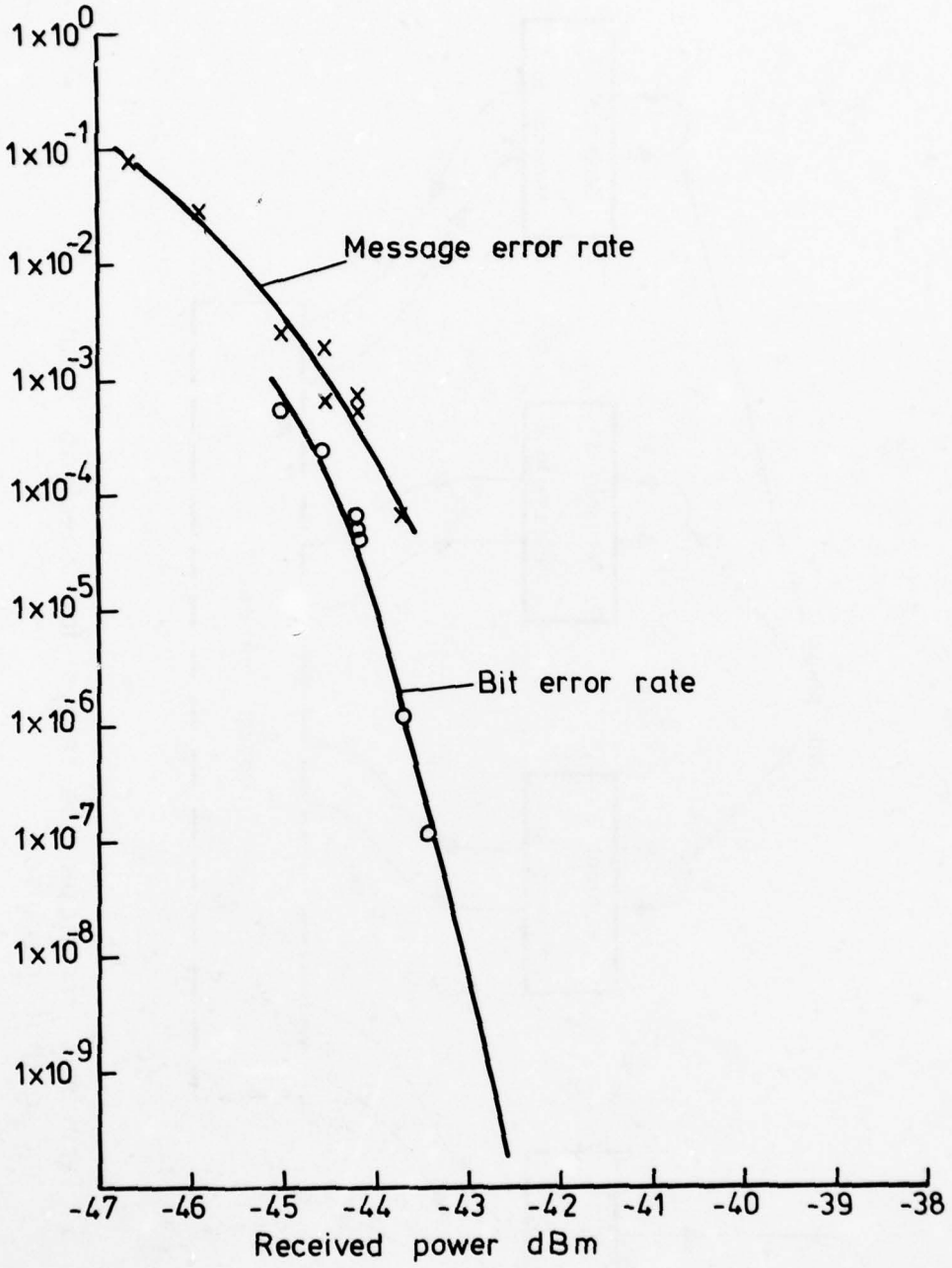
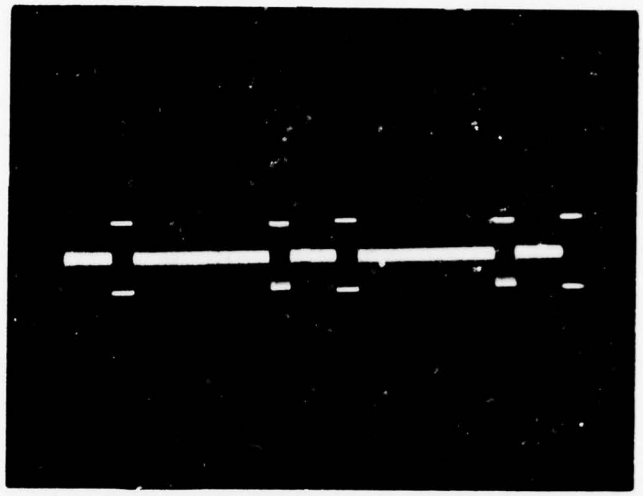
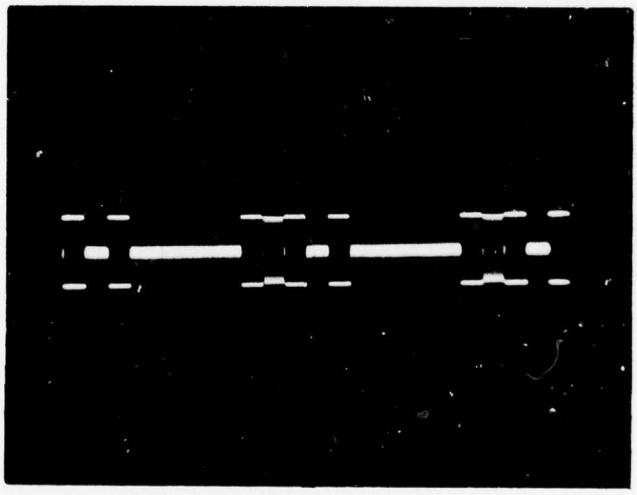



Fig.4.6 Error rate vs received power for one receiving terminal



(a) Test condition 1 - without extra transmitters



(b) Test condition 2 - with extra transmitters

Fig. 4.7 Optical receiver outputs during error tests

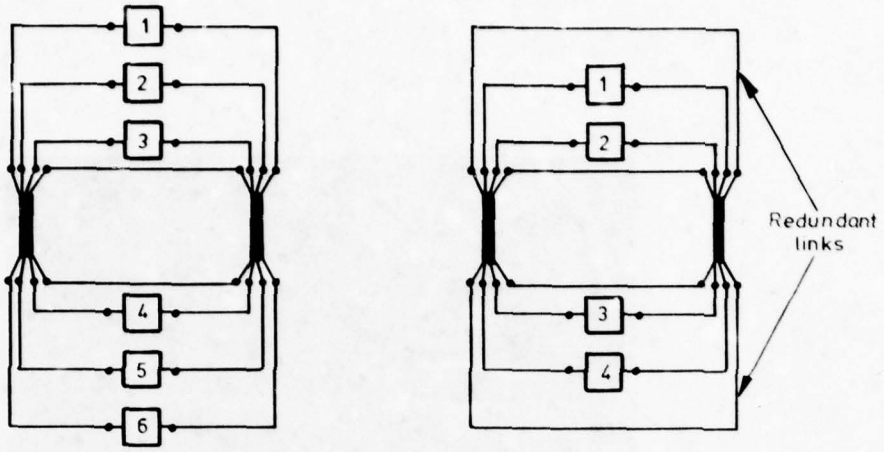



Fig.5.1 Harness configuration for up to six terminals employing two 4-way transmission stars

# THE IMPACT OF FIBER OPTIC MULTIPLEXING ON DISTRIBUTED AVIONICS ARCHITECTURE DESIGN

James M. Bain
Air Force Avionics Laboratory
Dayton, Ohio

## ABSTRACT

The development of low cost microcomputers has initiated a trend towards distributed architectures in avionics. Such architectures require high levels of interprocessor communications which may exceed the capacity of current 1 MHz avionic multiplex buses. The wide bandwidth (10 MHz) of a fiber optics multiplex bus can be used to satisfy the increased bus traffic requirements of distributed processor systems, however, this is not the only benefit of using a wideband bus in such systems. Due to the unique characteristics of distributed architectures, it is possible for the designer to use fiber optic's wide bandwidth to enhance system capabilities. The Air Force has developed a conceptual avionics architecture which demonstrates the advantages of using fiber optics multiplexing in a distributed processing environment.

## TRENDS IN AVIONICS PROCESSING ARCHITECTURES

Digital avionic architectures of the mid 1960's consisted of one or two central minicomputers interfaced to a large analog-to-digital (A/D) converter, which in turn was interconnected to various sensors, actuators, controls, and displays through a network of dedicated wire paths. Examples of this centralized processing approach include the F-111D Mark II system and the A-7D avionics.

In the F-15, a federated processing approach was taken in which separate minicomputers of different types are used to perform navigation, radar and electronic countermeasures processing.[1] Another innovation of the F-15

was to distribute A/D conversions throughout the aircraft in the form of Remote Terminals (RTs) interconnected by a serial digital multiplex data bus.

The recent appearance of low cost microprocessors has made it practical to distribute processing to the avionics subsystems themselves. This reduces the amount of data which must be bused to the minicomputer for processing and consequently reduces both data latency and bus loading. This "partially distributed" approach was used on the F-16 program.

For future avionics systems, it will be possible to go a step further and eliminate the minicomputer entirely, replacing it with a network of standardized, homogeneous single chip microcomputers embedded in avionics subsystems and interconnected by a data bus. Such designs are commonly called "distributed processing systems" and have the potential to provide many advantages, most notably, low cost, modular hardware, incremental growth, self organizing architectures,and simplified software.

In summary, a trend towards distributed architectures in avionics is apparent. Architectures have evolved from centralized to federated to the currently used partially distributed approach; the microcomputer revolution and the advantages of distributed networks assure that many future systems will be fully distributed.

## IMPACT OF DISTRIBUTED PROCESSING ON MULTIPLEX SYSTEM REQUIREMENTS

The distributed processing approach has a number of implications on multiplex system design. To explore these, the Air Force conducted the Distributed Processor/Memory Study to design a distributed architecture and associated multiplex system for avionics.[2,11]

A significant finding of the DP/M study was that a fully distributed processing approach generates more data bus traffic than a centralized or federated approach. This is the net result of two effects of distributed processing, one which tends to reduce bus traffic and another which tends to increase traffic. The effects are described below.

As mentioned before, distributed microprocessors can be used to reduce bus traffic in a centralized or federated system. If a data source and sink are located

at the same terminal, and if the amount of processing required is small, a local microprocessor at the terminal or subsystem can provide the necessary processing. Consequently, it will no longer be necessary to bus the data to a minicomputer for processing and bus processed data back to the terminal. This saves two bus transactions each time new data is processed. Note that this technique of using microprocessors to reduce bus traffic can only be used when processing results are not needed by other terminals or computers. Since avionics processing is characterized by large amounts of shared data, a point will be reached at which adding more microprocessors will no longer reduce bus traffic.

If, at this point, the minicomputer is eliminated and replaced by additional distributed microprocessors, the result will be a fully distributed processing system. Note that this transition involves the separating into multiple processors of software tasks which formerly resided in one processor and communicated via a shared memory. Once partitioned into separate processors, these software tasks must communicate over the bus, consequently, data bus traffic will increase. The degree of increase depends on the number of microcomputers used, their size, and the interprocess communication requirements of system software. Quantitative analyses indicate that if avionic system processing is partitioned into numerous small processors, the increased bus traffic requirements will be beyond the capabilities of current 1 Megabit per second (Mbps) avionic multiplex systems.[2,3*] Using MIL-STD-1553A protocol, the author has calculated that a 2-4 Mbps bus would be required for such systems.

## MULTIPLEX SYSTEM DESIGN FOR DISTRIBUTED PROCESSING NETWORKS

Having established that many future avionics processing schemes will be distributed and that bus traffic in some such systems will exceed the capabilities of current multiplex systems; the need to design high performance multiplex systems for distributed processor networks is apparent. The DP/M study developed such a system using standard 1 MHz capacity twisted shielded wire pair (TSP) multiplex channels. The approach was to increase available system bandwidth by using a multiple hierarchical bus topology, and to minimize bus traffic requirements by optimal partitioning of software. Since the bus design uses standard multiplex cable, it could be implemented in near-term systems.

* A 1MHz channel can carry 1Mbps of Manchester encoded digital data.

For future systems, an alternate approach to satisfying the increased bus traffic requirements of distributed processor networks will be to use a wideband fiber optics data bus. Recently, the Air Force has developed a laboratory demonstrator of a nine-terminal data bus which can execute MIL-STD-1553 bus protocol at a rate of 10 Megabits per second,[4] a tenfold increase in data throughput over current wire systems.

Aside from wider bandwidth, fiber optics cables differ from wire in that they are dielectric, unipolar, and carry very low power levels. These issues were addressed during the design of the fiber optics bus [4,5] and MIL-STD-1553 compatibility was achieved.

## IMPACT OF FIBER OPTICS MULTIPLEXING ON DISTRIBUTED AVIONIC ARCHITECTURE DESIGN

As described previously, the wide bandwidth of a fiber optics bus can be used to satisfy the increased bus traffic requirements of a distributed processor system. However, this is not the only benefit of using a fiber optics bus in a distributed system. Because of the unique characteristics of distributed architectures, it is possible for the designer to use fiber optic's additional bandwidth to increase system capabilities and/or simplify system integration. The following are five major areas in which fiber optics multiplexing impacts tradeoffs in the design of distributed avionics architectures.

First, a wideband bus allows increased interprocessor communications (IPC). Consequently, processing can be partitioned into numerous, single chip microcomputers without overloading the bus. Furthermore, software need not be partitioned to minimize bus loading, but can be partitioned to optimize other factors. In addition, increased IPC capacity makes possible the use of advanced distributed processing concepts such as dynamic task scheduling[6], resource sharing, and self organizing architectures.

Secondly, a wideband bus allows the use of message formats and protocols which are highly fault tolerant. For example, each message can contain explicit source, destination, sync, retry, and status codes in addition to data. To assure maximum integrity, each word can contain error detection/correction codes and each message can have some sort of error check at the end. Protocols can be used in which each bus transmission

is explicitly acknowledged by the receiver with a negative or positive response. Broadcasts can be replaced by a multitude of single destination messages with acknowledgements from each receiver.

Thirdly, a wideband data bus allows additional data sources to be placed on the multiplex system both during the original design, and in subsequent retrofits. A corollary of this is that higher frequency data sources can be placed on the bus. MIL-STD-1553A recommends exclusion of data sources above 3KHz bandwidth and calls those between 400 Hz-3KHz "areas of questionable application" for a TSP data bus.[7] On a 10 Mbps fiber optics bus, these limits can be increased significantly.

A fourth implication of a wideband system is that bandwidth can be used to simplify bus traffic management by reducing requirements for bus "efficiency" and apriori bus traffic scheduling. Studies in software development have shown that if a computer program must be written to operate in a constrained address space, the cost of programming is increased significantly.[8] An analogous situation exists when digital data must be transferred through a channel of constrained bandwidth. Various measures are employed to increase data bus "efficiency" and thereby squeeze additional data onto the bus. With a tenfold increase in available bandwidth, "efficiency" is no longer important. In fact, using such advanced concepts as bus contention and self-organizing architectures may allow the elimination of apriori bus traffic definitions altogether, greatly simplifying system integration and retrofit.[9]

The fifth, and final implication of a wideband system to be discussed here is that of transaction timing. Data latency, response time and system reconfiguration time can potentially be reduced using a wideband bus and contention access algorithms.

Examination of the tradeoffs discussed above reveals that in certain cases the combination of a fiber optics data bus and distributed microprocessors in an avionic environment tends to be synergistic: fiber optic's bandwidth can be used to obtain significant architectural advantages, but intelligent BIUs*are required to use this bandwidth; fully distributed processing can provide advantages, but requires high communication bandwidth.

---

* Bus Interface Units ( also called "terminals" )

For example, an optical bus provides bandwidth required for such schemes as dynamic task scheduling and error correcting codes, but distributed processors are required to implement such schemes; conversely, distributed architectures provide the potential to partition software among low cost single-chip processors, but this generates additional bus traffic. It could be said that a distributed network is needed to make optimal use of an optical bus, and vice versa. The combination of distributed processing and fiber optics has the potential to provide highly advanced architectures.

## A CONCEPTUAL ARCHITECTURE FOR AVIONICS

The previous section listed some of the ways in which the capabilities of fiber optics multiplexing could impact the design of distributed architectures. The designer of a future avionics system must select from this list those techniques which are applicable to his system requirements. He must keep in mind that some of the techniques described are mutually exclusive and others are still in the research stage. Nevertheless, the emerging technologies of fiber optics and distributed processing give the designer a vast array of architectural alternatives.

As a first step in realizing the potential advantages of fiber optics and distributed processing, the Air Force Avionics Lab has conducted a study to define a distributed avionics architecture concept with wideband fiber optics multiplexing.[3] The approach was to select a specific avionics suite and mission, and to design a system to satisfy those requirements by selecting those techniques described in the previous section which were felt to be most applicable.

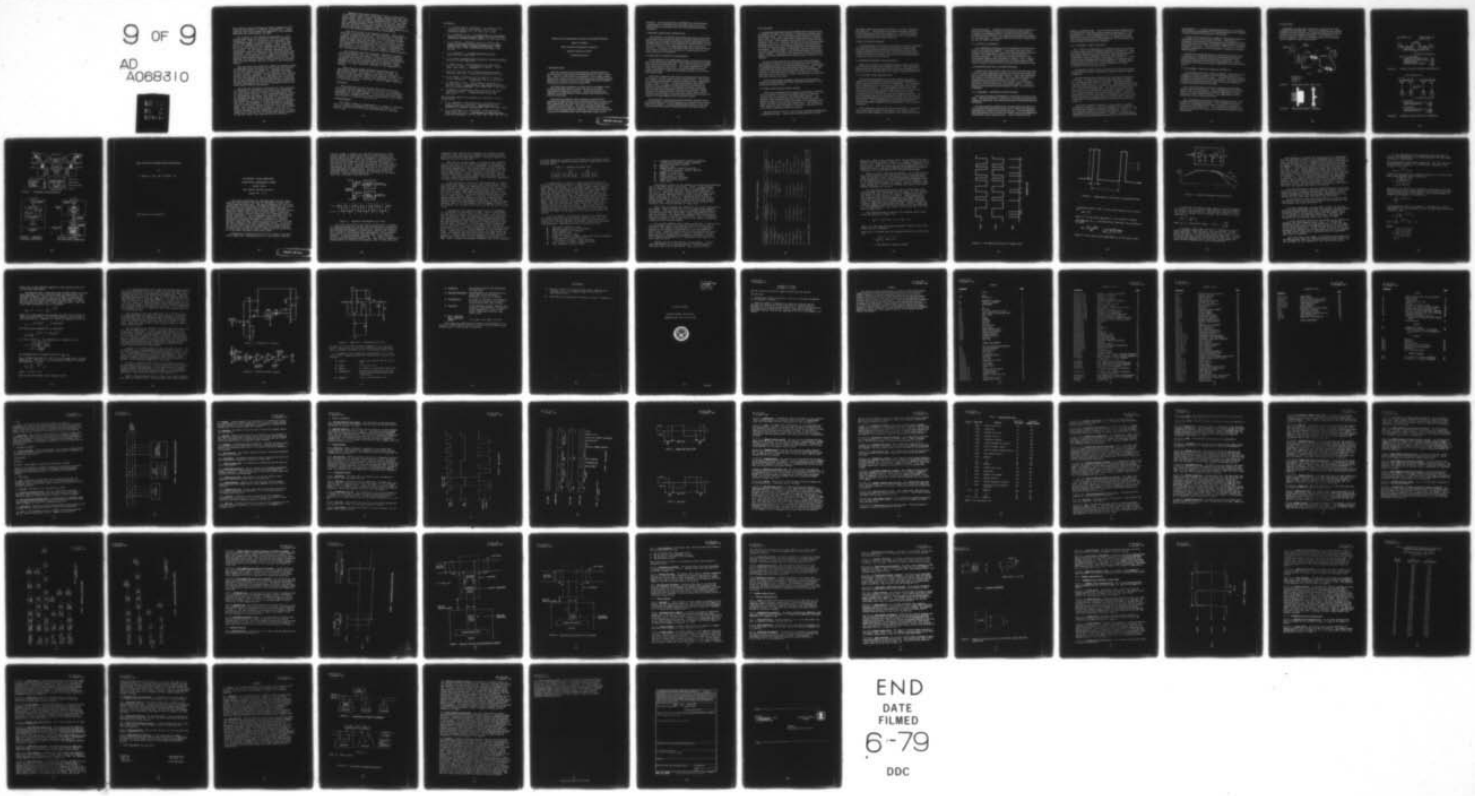
Three candidate avionic distributed processing architectures were developed during the study and are described in detail in reference 3. One was optimized for performance,another for fault tolerance, and the third for low development risk. A fault tolerant architecture called the Passive Bus System will be briefly described here. The description will emphasize topology/redundancy, bus control, protocol, message formats, and subsystem software interface. Detailed issues of hardware implementation were beyond the scope of the study.

The system topology features a quad redundant bus architecture and a passive fiber optics bus. Radial

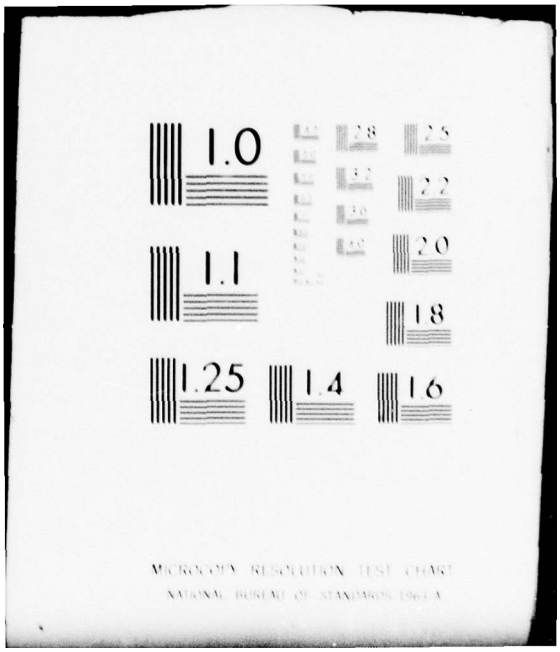
AD-A068 310 AERONAUTICAL SYSTEMS DIV WRIGHT-PATTERSON AFB OH F/G 17/2
PROCEEDINGS OF THE AFSC MULTIPLEX DATA BUS CONFERENCE (2ND), HE--E
OCT 78 E C GANGL

UNCLASSIFIED ASD-TR-78-34-VOL-2 NL

9 OF 9
AD A068310



END
DATE FILMED
6-79
DDC


1.0
1.1
1.25
1.4
1.6
2.0
2.2
2.5
2.8
3.2
3.6
4.0
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

arm couplers would be located at widely separated locations on the aircraft to enhance survivability. Fully redundant BIUs were judged too expensive; non redundant BIUs were not sufficiently reliable.

Bus control in this system is performed in a contention mode using an "Ethernet"[10] form of contention. Consequently, there is no central bus controller and no computer memory is needed to store static bus schedules. The scheme works as follows: A Bus Interface Unit, when desiring to send a message, waits until the bus is quiet and begins to transmit, and "listens" for a collision. If one occurs, the BIU transmits invalid modulation briefly to make sure that all BIUs detect the collision. After the collision, each BIU refrains from using the bus (times-out) for a random time period, with standard deviations adjusted according to priority. The BIU with the shortest time-out gains control of the bus. When that BIU vacates the bus it sends a "bus available" signal.

The collision bus allocation scheme was chosen because it exhibits a high tolerance to errors and BIU failures (no reconfiguration is required). Resynchronization occurs with each transaction. In addition, the scheme is fast, dynamic, provides reduced response time, and has optimal priority response. It satisfies the requirement of real-time systems to assign opportunities to use the bus in any sequence and with any percentage of bandwidth. The collision scheme is well-matched to bus systems with minimum propagation delay and wide bandwidth such as the fiber optics Passive Bus System.

The protocol selected is designed for high integrity. All messages in the passive bus system are transmitted on a one-to-one basis from source to destination with no third party involved. This single-destination method of transmission was chosen over broadcast mode because it provides improved fault tolerance at the expense of bus bandwidth. This is an acceptable tradeoff because with fiber optics there is little motivation to minimize the use of bus bandwidth. Single-destination messages allow for definitive positive and negative acknowledgements, thus raising system integrity. With four 10 Mbps buses, most performance requirements can be met with virtual broadcast at the subsystem level which is translated into a multiplicity of single-destination messages at the BIU level, thus maintaining high integrity.

Message formats were designed to carry a large amount of information including source name, destination name, word count, error codes, a cyclic redundancy check code, failure notification, retry inhibit, synchronization, and of course, data. Given this vast amount of data on the bus, each BIU has a detailed picture of the current and past bus messages and can use this information to detect errors/faults and devise a strategy to overcome them. The 10 Mbps fiber optics bus provides ample bandwidth for this information.

The error coding is of special interest. Each 22 bit word has a 5 bit error code, and each message is followed by a cyclic redundancy check code. The fiber optics bus was assumed to have a bit error rate of 1 in $10^8$ since hardware which achieves this rate has been built (reference 4, 12). Based on these assumptions, the use of error coding reduced the probability of an undetected bit error in the Passive Bus System to 1 in $2.6 \times 10^{19}$ bits. Practical considerations may preclude the achievement of this number in practice, but the potential certainly exists to surpass current systems.

The subsystem software interface consists of a queue containing message data, not pointers. The queue operates in a first-in/first-out manner. A high-priority mechanism is unnecessary because of the high fiber optic bus bandwidth. Similarly, there is no message removal mechanism because messages will be sent before they can be removed. Consequently, the queue controller hardware is not expensive or complex.

Space does not permit detailed discussions here of quantitative design tradeoffs which formed the basis for the system design. The reader is referred to reference 3 which contains excellent quantitative analyses of such topics as error coding and bus traffic, and describes how these findings were used in a systematic "solution space" approach to the design process.

## CONCLUSION

The conclusion is that distributed architectures are here, fiber optics buses are coming, and that it is feasible to combine the two technologies to obtain cost and performance advantages for avionics systems. It is hoped that this paper will provide a perspective of these technologies, and of their implications in avionics system design.

## ACKNOWLEDGEMENT

The author wishes to acknowledge E. D. Jensen, G. Marshall, and J. A. White of Honeywell who designed the Passive Bus System, and whose writings were abstracted here to describe that system.

REFERENCES


1. J. F. Mason and M. Shunfenthal, "In Avionics, Distributed Processing Has Become a Way of Life," Electronic Design, Oct. 25, 1976, pp. 74-78.

2. M. D. Johnson, et al., All Semiconductor Distributed Aerospace Processor/Memory Study, Air Force Avionics Laboratory Technical Report AFAL-TR-73-226, Aug. 1973.

3. E. D. Jensen, G. Marshall, and J. A. White, The Impact of Wideband Multiplex Concepts on Microprocessor-Based Avionic System Architectures, AFAL-TR-78-4, Feb. 1978; Available as DDC Report #AD-A057878.

4. J. E. Shaunfield, EMI/EMP Resistant Data Bus, AFAL-TR-76-99, Sept. 1976.

5. J. R. Biard, Optoelectronic Aspects of Avionic Systems, AFAL-TR-73-164, June 1973.

6. C. Wise, et al., "Multiprocessing for Electronic Warfare Avionics," Proceedings of NAECON, Dayton, Ohio, May 1978, pp. 137-141.

7. "Aircraft Internal Time Division Command/Response Multiplex Data Bus," MIL-STD-1553A, April 30, 1975.

8. B. W. Boehm, "Software and Its Impact: A Quantitative Assessment," Datamation, May 1973, pp. 48-59.

9. C. Wise and J. Bain, "Distributed Processing for MIL-STD-1553A," Proceedings of the Second AFSC Multiplex Data Bus Conference, Oct. 1978, Vol. 1, pp. 51-58.

10. R. Metcalf, "Distributed Packet Switching for Local Computer Networks," Communications of the ACM, July 1976.


The following papers are synopses of references 2 and 4, respectively.


11. G. A. Anderson, "Interconnecting a Distributed Processor System for Avionics," Proceedings of the Symposium on Computer Architecture, Dec. 1973, pp. 11-20.

12. J. R. Biard and J. E. Shaunfield, "A MIL-STD-1553 Fiber Optic Data Bus," Proceedings of AFSC Multiplex Data Bus Conference, Dayton, Ohio, Nov. 1976, pp. 177-236.

# FIBRE OPTIC REGENERATIVE MULTI-ACCESS NETWORK

Robin D. Beasley

Flight Automation Research Laboratory

Marconi Avionics Limited

Rochester Kent UK

## 1 INTRODUCTION

Marconi Avionics Limited recognised the potential benefits to avionic systems of fibre optic data transmission more than a decade ago. After two initial research programmes, the company implemented the use of fibre optics on the YC14 flight control system and also entered into a Ministry of Defence (PE) funded fibre optic component development programme known as MINILINKS.

The MINILINKS components were then used in installation trials at British Aerospace Warton, which were carried out under representative production conditions. These programmes have resulted in A-B links based on fibre bundles being established as a practical technology.

The advent and apparent rapid acceptance of Mil-Std-1553 combined with other avionic and aviation developments, has led to the likely requirement for a fibre optic multi-access network which is compatible with the Mil Std. The Flight Automation Research Laboratory (FARL) of Marconi Avionics Ltd undertook the development of such a network twelve months ago, and the result of this work is known as FOREMAN (Fibre Optic REgenerative Multi Access



PRECEDING PAGE BLANK

Network). The main objective of FOREMAN is to allow the rapid implementation of a Mil-Std-1553 fibre optic network, but with minimal effect on existing avionic unit construction and airframe installation.

## 2 EXISTING FIBRE OPTIC TECHNOLOGY

The problems of using fibre optic bundle technology in the avionic environment were overcome by Marconi Avionics Limited during the design of the YC14 Flight Control System and the MINILINKS contract. The MINILINKS components were designed for general system use and thus the technology is very applicable to fibre optic multi access networks. A brief review of the MINILINKS contract and the installation trials is included in this section as it relates directly to the design philosophy of FOREMAN.

### 2.1 MINILINK COMPONENT DEVELOPMENT

At the beginning of the MINILINKS contract, a survey was carried out to identify the most applicable system specifications for an avionic fibre optic data transmission link. Fig 1 shows a pictorial representation of the results of the survey. The survey identified four main areas of development; these were, military standard terminals, avionic connectors, a shop floor termination technique and an avionic fibre bundle cable.

Fibre optic transmitter and receiver terminals were developed in three stages, and Fig 2 shows the final hybrid module. This module is a military standard hybrid capable of operating over a temperature range of +125 $^{o}$C to -55 $^{o}$C. An important feature of this hybrid module is that it is capable of being mounted on a printed circuit board in a similar manner to conventional electronic components. The terminals are a specifiable interface; the transmitter module accepts a TTL signal and emits a specified optical output, and the receiver module accepts a minimum optical power and reproduces the original TTL signal applied to the transmitter module.

It was decided to use existing avionic electrical connector shells to avoid the delay involved with approval or re-approval for an avionic fibre optic connector. During the MINILINKS contract a fibre optic rack mounted DPX connector and hand mated TYPE 602 connector

were developed.

An important development during the MINILINKS contract was the formulation, by FARL, of a 'shop floor' termination technique known as the 'Hot Crimp'. The 'Hot Crimp' process is based on the fibre optic bundle, contained by a glass bead, being pushed into a tapered hole in a hot jig. This results in the end face of the bundle forming a hexagonal close pack. The basic elements for this process have been designed to form a size 12 ferrule on completion of the termination. The 'Hot Crimp' has a number of advantages over conventional epoxy termination. The formation of a hexagonal close pack reduces the connector attenuation by 30%, it also forms an all-glass face which is easier to polish. The quality of the termination is controlled by the quality of the piece parts and thus requires no skill on the part of the operator.

The fibre bundle selected was made up of 61 fibres forming a bundle diameter of 800 μm and has an attenuation of 250-500 dB/km. The sheathing consists of two non-flammable plastics. A dual sheath was adopted with a soft inner sheath and a hard outer sheath. This construction was found to provide the fibre bundle with adequate protection, whilst allowing the standard 'hex crimp' tools used on conventional electrical cable to be used to mechanically terminate the fibre optic ferrule to the cable.

All the forementioned components have been environmentally tested at the Royal Signals and Radar Establishment Malvern, to the specification shown in Fig 1.

## 2.2 FIBRE OPTIC INSTALLATION TRIALS

A fibre optic installation trial, funded by the Ministry of Defence (PE) was carried out by British Aerospace Warton and Marconi Avionics after the completion of the MINILINKS contract. The purpose of this installation trial was to allow British Aerospace Warton to assess, from practical experience, the suitability of fibre bundle technology for avionic data transmission. The emphasis of these trials was on the installation of fibre optic components in the most realistic production conditions possible.

Marconi Avionics procured a kit of parts consisting of MINILINKS components and tools. These were then demonstrated to British

Aerospace production personnel consisting of wiremen, supervisors and inspectors. This production team then wired up an airframe metal mockup without scientific supervision. These trials proved that the installation of fibre optic bundle components constituted no major technical risk to any future projects.

## 3 MULTI ACCESS NETWORKS

Theoretical fibre optic network configurations have been expounded for a number of years. The advent of a widely accepted specific electrical highway standard (Mil-Std-1553) and the solution of the engineering problems associated with A-B fibre links has led FARL to the formulation of a solution to allow the practical implementation of fibre optic multi access networks.

### 3.1 EXISTING NETWORK CONFIGURATIONS

There are a number of configurations being proposed at present, most if not all fall into the two categories below. The following text assumes that the maximum transmitter to receiver attenuation for a practical,military standard,fibre optic system is approximately 32 dB.

#### 3.1.1 CENTRAL STAR DISTRIBUTION

The central star distributor is a system whereby the optical output from one transmitter is optically split and distributed to the other network terminals by means of a central star. Fig 3 shows the basic configuration for two terminals in a practical avionic 32 way central star distributor. Two de-mateable connectors have been provided at each system Line Replaceable Unit (LRU) to allow normal first and second line maintenance. The central star is shown contained in an LRU with the necessary de-mateable connectors. An in line bulkhead connector has been placed in each arm of the star as this would undoubtedly be necessary in an aircraft with more than one avionics bay. This configuration would result in a transmitter to receiver attenuation of greater than 50 dB and hence is not a practical proposition.

This configuration also has two other disadvantages. The central star could produce a common mode failure, and it also results in unnecessary cable lengths, as two adjacent LRUs would communicate

via the central star. A central star distributor system of about ten terminals is feasible in the avionic situation but would require a concession on installation practice, as the central star and all the airframe wiring would have to be supplied and installed as a preformed, one-piece loom. This type of network would only be appropriate for use in a single avionic bay.

### 3.1.2 OPTICAL T HIGHWAY

This configuration would appear to be suitable as it has the same format as the electrical highway. Fig 4 shows that when all the connectors necessary for first and second line maintenance have been included, this configuration results in very high transmitter to receiver attenuation. The main factor in this attenuation is the accumulated insertion loss of successive T pieces. This configuration has the additional disadvantage that its implementation would require an optical receiver with a very high optical attenuation range.

## 3.2 REGENERATIVE MULTI ACCESS NETWORK

Conventional forms of fibre optic multi access network were found to have too many fundamental problems to allow the rapid implementation of a Mil-Std-1553 fibre optic network. FARL have approached the problem in a different manner. The only other alternative, as networks that involve passive distribution result in too high an attenuation between transmitters and receivers, is to incorporate some form of active regeneration of the optical signal. A network based on this principle has been designed at FARL, and is known as FOREMAN.

### 3.2.1 FOREMAN - EXTERNAL CONFIGURATION

The external cabling of FOREMAN is restricted to the use of A-B links. This gives maximum utilisation of component development and installation experience, thus greatly assisting in reducing development timescales.

The only method of implementing a network using A-B links is to form a daisy chain where each avionic unit is linked to its immediate neighbours by a single A-B bi-directional link. The disadvantage of this configuration is that a failure of a single cable or any regeneration terminal will cause a system failure. This is overcome by the

addition of redundant links. The currently preferred configuration is for terminal N to have direct optical access to terminals $N \pm 1$, and terminals $N \pm 2$. This configuration will allow the survival of two cable failures and/or one terminal failure. It should be noted that this is the minimum failure survival capability, as more failures can be survived provided they are not adjacent.

### 3.2.2 FOREMAN - BASIC PRINCIPLE

Fig. 5 shows five terminals of a FOREMAN system, and terminal N shows the basic internal elements required to implement it. This particular configuration of FOREMAN entails any terminal having an optical path to the four adjacent terminals, thus terminal N is connected by a fibre optic cable to terminals $N \pm 1$ and $N \pm 2$. It can be seen from the diagram that the four cables from terminals $N \pm 1$ and $N \pm 2$ enter terminal N by a 4-way connector. The four cables are then joined by a four way Y, the main stem of the Y being connected to a bi-directional terminal.

An important feature to note is that each LRU only has one active terminal and thus external redundancy is gained by the use of passive components.

This network has been designed for the transmission of pulses of fixed width. Consider the case of the optical emitter in terminal N - 2 producing a light pulse of fixed width. This pulse would be relayed to terminals N, N-1, N-3 and N-4. The following effect which will be described in detail in relation to terminal N occurs in terminals N-1, N-3 and N-4 simultaneously. The pulse of fixed width from terminal N-2 enters terminal N via the 4-way connector. The four way Y routes this pulse to the bi-directional terminal. When the optical detector receives this pulse, the terminal circuitry will cause the optical emitter to produce a pulse of similar width. The 4-way Y will cause this regenerated pulse to be conveyed to terminals $N \pm 1$ and $N \pm 2$.

The pulse conveyed to terminal N - 2 has no effect; the pulse conveyed to terminal N - 1 forms the redundant link, complementing the optical pulse transferred between terminals N - 2 and N - 1. The two remaining signals are conveyed to terminals N + 1 and N + 2. These signals will cause terminals N + 1 and N + 2 to be activated by terminal N in the same manner as terminals N-1 and N when activated

by terminal N - 2. Thus the regeneration process is re-iterated as many times as is necessary for the original light pulse to be conveyed to all the terminals (32 maximum) in the network.

A multi-access network that is capable of sending short pulses of fixed width to up to 32 terminals now exists. This design is compatible with existing technology and aircraft installation practices. Fig 5 shows that the minimum FOREMAN configuration has an optical attenuation of 20.5 dB. This leaves an 11.5 dB margin on the 32 dB maximum transmitter to receiver attenuation allowable in a military standard system.

Mil-Std-1553 utilises Manchester encoding which has led to the development of a modulation technique which allows a Manchester encoded signal to be relayed using pulses of fixed width. This modulation technique also allows the bi-phase signal (3rd state) to be encoded, thus achieving complete compatibility with a Mil-Std-1553 system.

### 3.2.3 FOREMAN - REMOTE TERMINAL INTERFACE

The FOREMAN terminal contains the analogue, digital and optical components required to replace the electrical data transmission components in a Mil-Std-1553 system. The point at which FOREMAN interfaces with the Remote Terminal is flexible. The implementation of FOREMAN would automatically replace the data bus cabling, the coupling transformer and the stub cable.

Fig 6 shows the optimum method of interfacing the FOREMAN system with a remote terminal. The FOREMAN terminal interfaces directly with such components as the Harris Manchester Encoder Chip or any future 1553 LSI chip sets, thus replacing the isolation transformer and the transformer driver.

Fig 7 illustrates an alternative method of interfacing which could be useful for development purposes. The FOREMAN terminal is located in a small additional LRU and is connected to the isolation transformer in the system LRU. This would allow FOREMAN to be used on an electrical 1553 system with minimal modification to existing hardware.

## 4 CONCLUSION

FOREMAN is a fibre optic multi access network which is fully compatible with Mil-Std-1553. The use of FOREMAN only requires a change in the data transmission components leaving all the remote terminal logic and subsystem unchanged. The maximum use of existing technology, results in FOREMAN being a low risk system which is capable of rapid implementation.

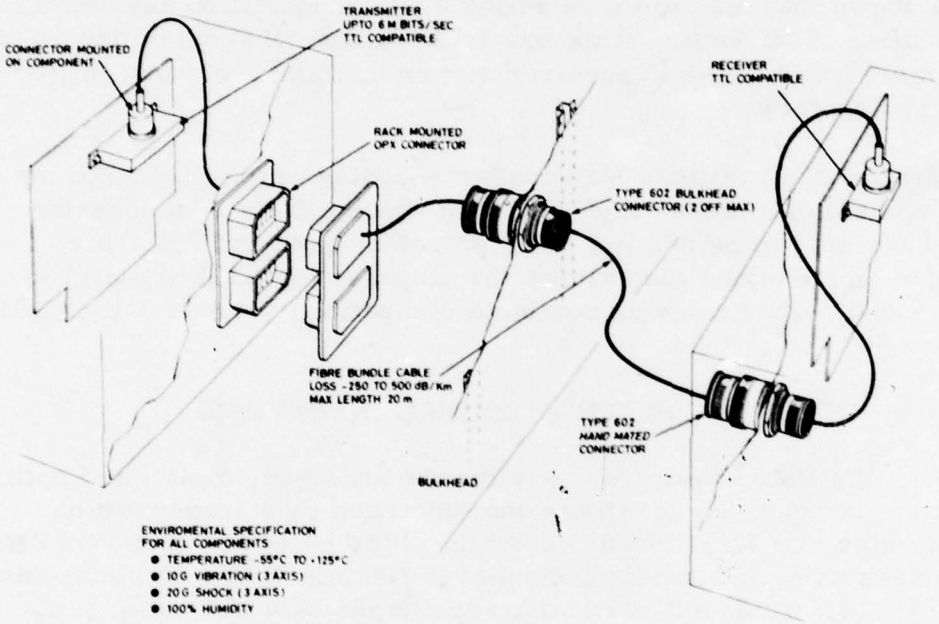



FIGURE 1 MINILINKS FIBRE OPTIC LINK

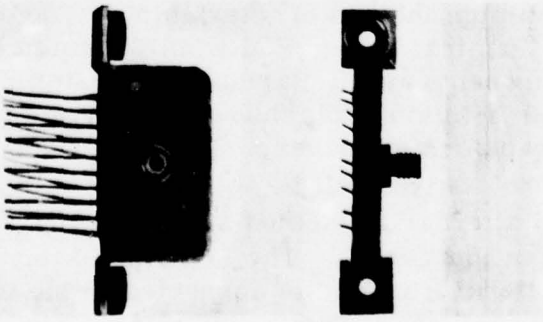
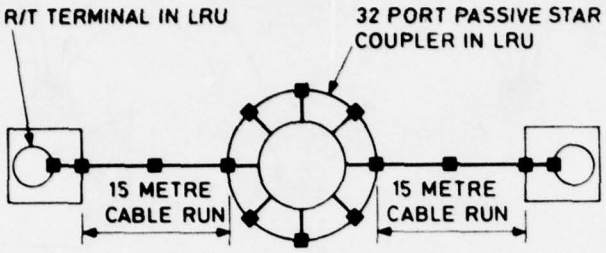


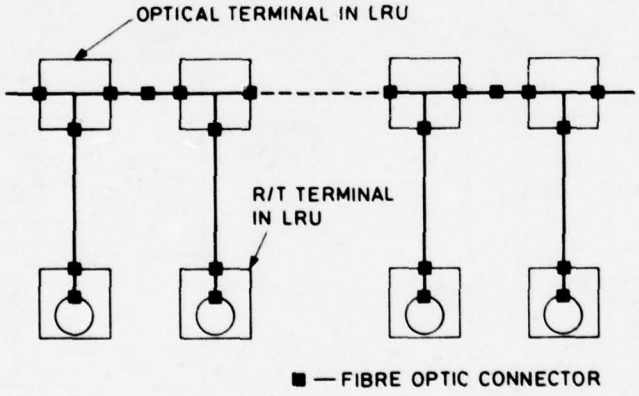
FIGURE 2 MINILINKS HYBRID TERMINAL




TOTAL R/T-R/T ATTENUATION

| | |
|---|---|
| THEORETICAL ATTENUATION OF STAR | 15dB |
| INSERTION LOSS OF STAR (ESTIMATED) | 6dB |
| CONNECTORS (8 OFF) | 24dB |
| 30 METRES OF CABLE (400dB/km) | 12dB |
| | ≤57dB |

FIGURE 3 AVIONIC 32-WAY CENTRAL DISTRIBUTOR




1st TO 2nd STATION

| | |
|---|---|
| THEORETICAL ATTENUATION R/T-R/T | ≈18dB |
| LINK ATTENUATION R/T-R/T | ≈27dB |
| | ≤45dB |

1st TO 32nd STATION

| | |
|---|---|
| THEORETICAL ATTENUATION R/T-R/T | ≈22dB |
| LINK ATTENUATION R/T-R/T | >100dB |
| | ≤>>100dB |

FIGURE 4 AVIONIC 32-WAY OPTICAL-T HIGHWAY

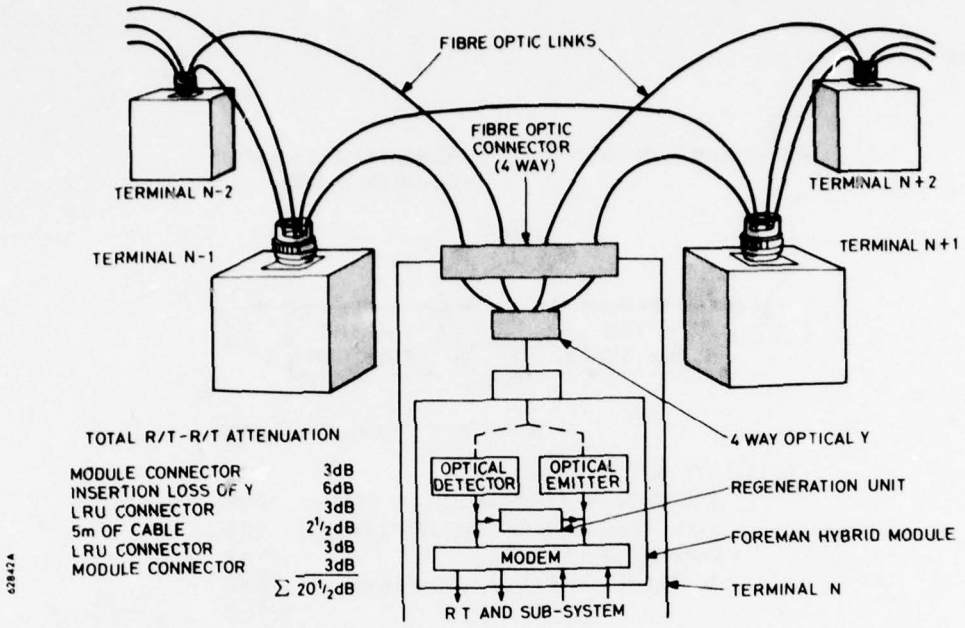



FIGURE 5 FOREMAN CONFIGURATION

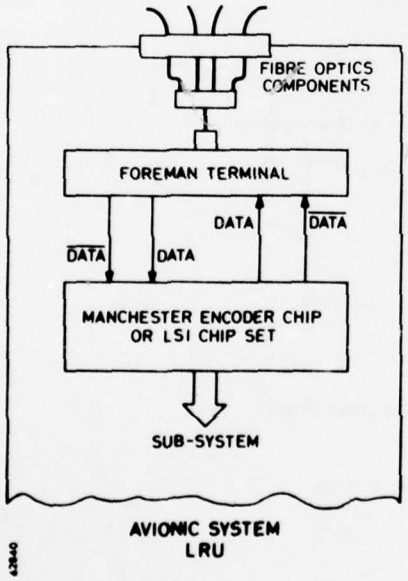



FIGURE 6 FOREMAN INTEGRAL INTERFACE

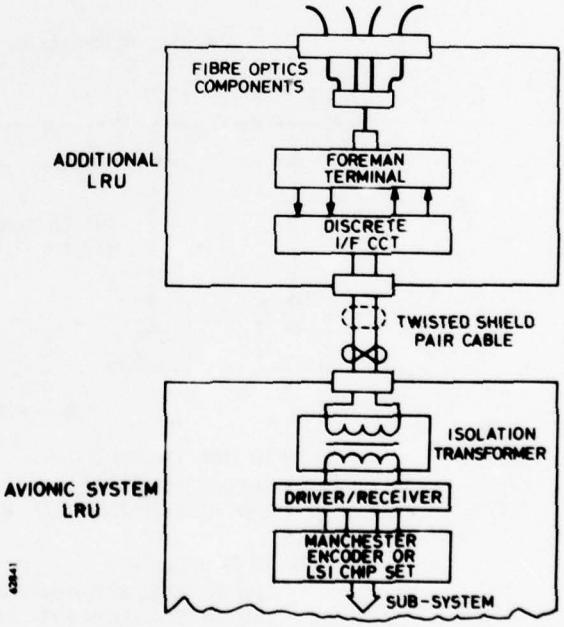



FIGURE 7 FOREMAN - NON-INTEGRAL INTERFACE

# FIBER OPTIC DATA BUS SYSTEM DESIGN CONSIDERATIONS

by

L. Faltzer, S. Cole, and R. Martinet, ITT

(This paper was not submitted)

# AN AIRCRAFT 1553B COMPATIBLE FIBER OPTIC INTERCONNECT SYSTEM

Robert Betts

IBM Federal Systems Division

Owego, NY 13827

This paper describes the tradeoffs made and the components and configuration selected to implement a fiber optic 1553B-compatible data bus. The basic assumption of near-term application and MIL-E-5400 environments restricted the choice of components and technology to that presently available. Among the system elements examined and traded off were sources, detectors, fiber type, single or bundle cable, connectors, optical couplers, optical waveforms, and transmitter and receiver electronics. To be compatible with MIL-STD-1553B required selection of some "standard" electrical interface available in existing equipment to minimize any required retrofit modifications. Such an interface is where the data has been encoded in two-level Manchester format. A conclusion of the studies was that a pulse position modulated optical waveform can improve significantly the system's signal-to-noise ratio and can be used to improve the bit error rate or to increase allowable worst-case losses.

Examination of MIL-STD-1553A and its imminent successor 1553B, shows that implementation of a fiber optic interson-



PRECEDING PAGE BLANK

nection scheme to replace the twisted shielded pair transmission medium, without major effect on the standard, requires making the fiber optics transparent to the protocol and bus architecture. This can be done by selecting the proper electrical interface in the transmitter and receiver electronics such that the system is unaware of whether the bus is wire or glass. Such an interface is where the digital information put on or received from the bus is a two-logic-level Manchester-encoded signal. Figure 1 is a block diagram showing both interfaces and the required electrical waveform. This interface exists or will exist in any equipment that conforms to MIL-STD-1553B.

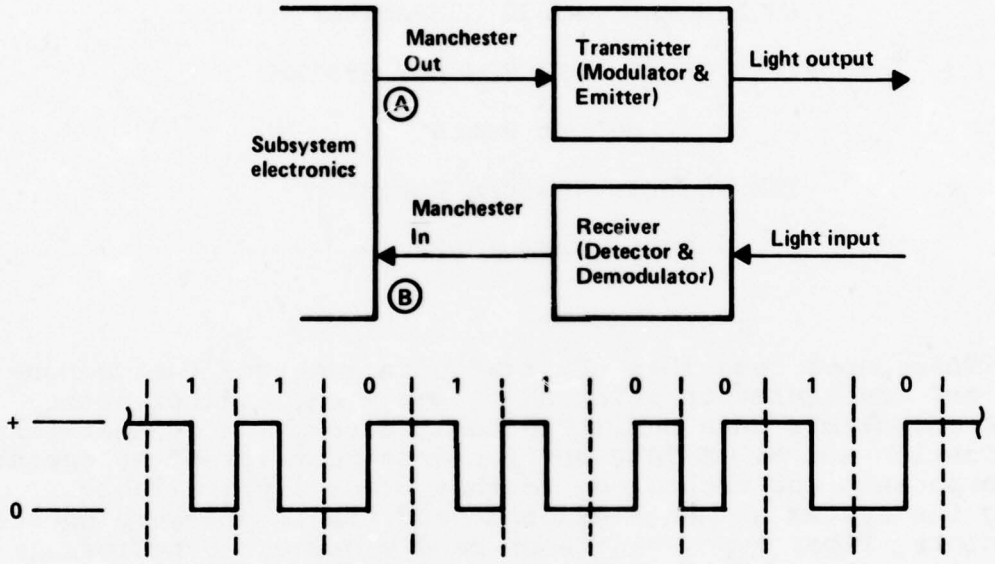



Figure 1. Two-Level Manchester In or Out

The first priority in the initial studies was to survey the existing market for optical and electro-optical components or technologies which were qualified or could be qualified to the military environment; in particular, meeting the temperature, humidity, and reliability requirements. This narrowed the field to large physical aperture LEDs (light-emitting diodes). Lasers have three problems; (1) their lifetime has not been sufficiently proven, (2) they require temperature stabilization in military environments, and (3) they are only

compatible with single-fiber systems, and location of the extremely small emitting area requires a pigtailed package which as yet has not been developed to the point where true hermeticity has been achieved.

LEDs come in various forms. To minimize the coupling loss at the input to the fiber bus system, an optically enhanced device (use of a lens or reflector in the LED chip packaging) appeared to offer significantly better coupling than nonenhanced types. Having selected the emitter of the optical bus system, the next task was to determine the optimum combinations of cable, connectors, couplers, and detectors. The "best" single-element selection does not necessarily imply the optimum system. It is the best combination of elements that must be selected.

A logical place to start is to establish what is called the system loss budget in view of other requirements. MIL-STD-1553A and B require an undetected error rate of $1 \times 10^{-12}$. This requires a good signal-to-noise ratio at the output of the receiver electronics, where the decision for a "1" or "0" is to be made. MIL-STD-1553A and B also require a bus with up to 32 ports. It quickly becomes obvious that a radial or "star" optical coupler is required to minimize the system losses without resorting to active coupling means that would detract from system reliability. In the star coupler, there is an unavoidable power division loss in power decibels (dB) of 10 log P where P is the number of ports in the system. For 32 ports this loss is 15 dB and for 16 ports, 12 dB.

A convention for comparing optical receiver capability is to describe its sensitivity in dBm. Plus or minus dBms are dBs above or below 1 mW of optical power. Two types of optical detectors have the speed and sensitivity required for this application. They are the silicon PIN photodiode and the silicon avalanche photodiode (PIN and APD respectively). The PIN diode has a limit of around -43 dBm sensitivity and the APD about -65 dBm for a one-to-one signal-to-rms-noise ratio. For a $10^{-12}$ bit error rate (BER), one requires approximately 14.5 to 1 signal-to-rms-noise level at the decision point for "1s and "0s". This translates to 11.6 dB above the noise. Adding this to the diode limits, one must supply at least -31.4 dBm to the PIN receiver and -53.4 dBm to the APD receiver. A good figure for the output of an optically enhanced LED, taking into account temperature and aging is -3 dBm. Table 1 shows the maximum allowable loss in the remainder

of a bus system for 16 ports and 32 ports for a PIN and an APD detector (detector sensitivity minus emitter power minus coupler division loss).

Table 1. Maximum Allowable Loss

| | PIN | APD |
|---|---|---|
| 16 ports | 16.4 dB (13.4) | 38.4 dB (35.4) |
| 32 ports | 13.4 dB (10.4) | 35.4 dB (32.4) |

In todays state of the art, there is an additional insertion loss in star couplers of approximately 3 dB due to nonuniformity, reflections, etc. If this is subtracted from the above table, the figures in parenthesis apply. These numbers, then, represent losses that are allowed for input and output coupling, connectors, and the fiber optic cable itself. It would appear that selection of the APD receiver was obvious, but other considerations such as temperature sensitivity and the high voltage (>200 V) required for operation of an APD would favor the PIN diode if it could satisfy the bus requirements. The next task was the tradeoffs between input and output coupling and connector loss against the cable configuration and optical fiber selection. Taking cost and requirements into account, and assuming the greatest distance between terminals on the bus as 100 meters, it was clear that fiber with losses in the range of 10 to 40 dB per km were required.

A set of computer programs were written for calculating the input coupling loss, the output coupling loss, the connector loss, and the system loss for best and worst case combinations of various emitters, detectors, cable, and connector configurations. Variables in these calculations were:

- LED source (aperture) size
- Radiation pattern (cosine exponent)
- Fiber diameter and tolerance
- Fiber bundle size
- Core-clad ratio of fiber
- Number of fibers in bundle (hexagonal pattern assumed)
- Source-bundle relative position
  (off-axis, rotation, angle, spacing)
- Core refractive index (reflection loss)
- Core numerical aperture (N/A)

- Connector bundle-bundle relative position (off-axis rotation, angle, spacing)
- Direction through connector
- Detector size
- Bundle-detector relative position (off-axis, rotation, angle, spacing)
- Number of connectors
- Number of ports on bus
- Output power of source
- Distance between terminals
- Cable end finish quality
- Cable loss.

Preliminary results of the computer calculations established two of the many choices to be made. First, it was obvious that to hold the input coupling loss to a reasonable value, the optically enhanced LED and a fiber bundle cable were imperative. The source physical aperture of available infrared (for speed) LEDs was around 1.2 mm, and the compatible bundle cable was 19 fibers in a hexagonal pattern with a fiber core diameter of 215 μm. Seven fibers at 360 μm was a possible choice, but for cable flexibility and bend radius, 19 was chosen. Thirty-seven fibers at 150 μm (another possibility, was rejected because of cable cost.

After many iterations and tradeoffs, the final configuration was chosen. Table 2 shows all of the variables and assumed tolerances, with the final typical, minimum, and maximum system losses along with the dynamic range requirements of the optical receiver. Not shown, but assumed, was a transmissive type star coupler to avoid the additional 3 dB in a reflective coupler required at the terminal end of the cable to separate transmitter and receiver paths.

For typical systems within these constraints, a PIN detector offers adequate margin for required signal-to-noise ratios. The worst case shown, although possible, is very improbable because of the required coincidence of all the elemental worst cases. However, deviations from typical most certainly will occur, and any decrease in loss or increase in optical power without a relatively corresponding increase in noise will enhance the probability of maintaining the required bit error rate of $10^{-12}$.

Various optical waveforms were investigated. A three-level Manchester as in the twisted-pair 1553A standard, a two-level zero-to-plus Manchester, and a pulse position

## Table 2. Final System Loss Analysis

| Input Coupling | Connector Coupling | Output Coupling | System Losses |
|---|---|---|---|
| Fiber diameter tolerance<br>±2.5% | Fiber diameter tolerance<br>±2.5% | Fiber diameter tolerance<br>±2.5% | LED output power<br>±5 dBm |
| Core index of reference<br>1.452 | Core index of reference<br>1.452 | Core index of reference<br>1.452 | Input coupling loss<br>5.1 dB |
| Fiber diameter/core:clad<br>215 μm/1:1 | Core/clad ratio<br>1:1 | Fiber diam./core:clad<br>215 μm/1.1 | Cable length (m)<br>0.001 0.01 0.1 |
| Source diameter<br>1.2 mm | | Chip diameter (PD)<br>1 mm | Cable loss (dB/km)<br>10 20 40 |
| Radiation Pattern<br>$\cos^{20}\theta$ | | | Connector loss<br>1.3 dB |
| Number of fibers<br>19 | Number of fibers<br>19 | Number of fibers<br>19 | Number of connectors<br>2 4 6 |
| Numerical aperature<br>0.3 | Numerical aperature<br>0.3 | Numerical aperature<br>0.3 | Number of ports<br>16 |
| Off-axis tolerance<br>*±2.5% | off-axis tol/rotation<br>*±2.5% | Off-axis tolerance<br>*±2.5% | Coupler excess loss (dB)<br>2 3 4 |
| Source/fiber separation<br>*5% | Bundle/bundle separation<br>*2.5% | Bundle/chip separation<br>*150% | Coupler nonuniformity (dB)<br>0 1 2 |
| Angular displacement<br>2.5° | Angular displacement<br>1.0° | Angular displacement<br>2.5° | Output coupling loss<br>3.2 dB |
| Lost 5.1 dB | Loss 1.3 dB | Loss 3.2 dB | Typical detector power - 29.6 dBm<br>Minimum detector power - 43.1 dBm<br>Maximum detector power - 19.9 dBm<br>Dynamic range - 23.2 dB |
| * % of fiber bundle diameter | | | |

modulation (PPM) easily encoded from, and decoded to the Manchester interfaces shown in Figure 1. Figure 2 shows the first few bits of a command word of the 1553A protocol encoded in the three forms. Both the three and two level Manchester were rejected in favor of the pulse modulation because of the gains achievable to be described in the following paragraphs.

There are three potential advantages; (1) a gain in signal-to-noise ratio of the square root of the relative amplitudes of the short pulse to a conventional average 50% duty cycle Manchester; (2) a constant non-variable threshold level at the decision point for "1s" and "0s" in the receiver, and (3) a simple way to handle the dynamic range of detector input power caused by differences in highest and lowest loss paths between terminals on the bus.

For the following derivations and for comparison with other systems, two assumptions will be made; (1) the principal source of noise is in the receiver preamplifier, and (2) the noise is white gaussian in nature. These assumptions are compatible with previous papers on the subject and are also real in terms of the systems and bandwidths involved. Figure 3 shows the type of pulse examined along with definitions of amplitude and normalized time. Figure 4 shows the receiver bandpass characteristics along with a low-frequency, $f_L$, and high-frequency, $f_H$, cutoff at the 3 dB and 3 or 6 dB point respectively.

The auto-correlation function for Gaussian white noise after the receiver "filter" is:

$$R_o(t) = \emptyset \int_o^\infty h(u) \cdot h\,(u + |t|\,)\, du$$

where h is the receiver impulse response, and $\emptyset$ is the noise power per unit frequency.

From this it follows that the standard deviation in rms noise is:

$$\sigma = \sqrt{\emptyset \int_o^\infty [h(u)]^2\, du}$$

$\sigma$ = rms noise at receiver output.

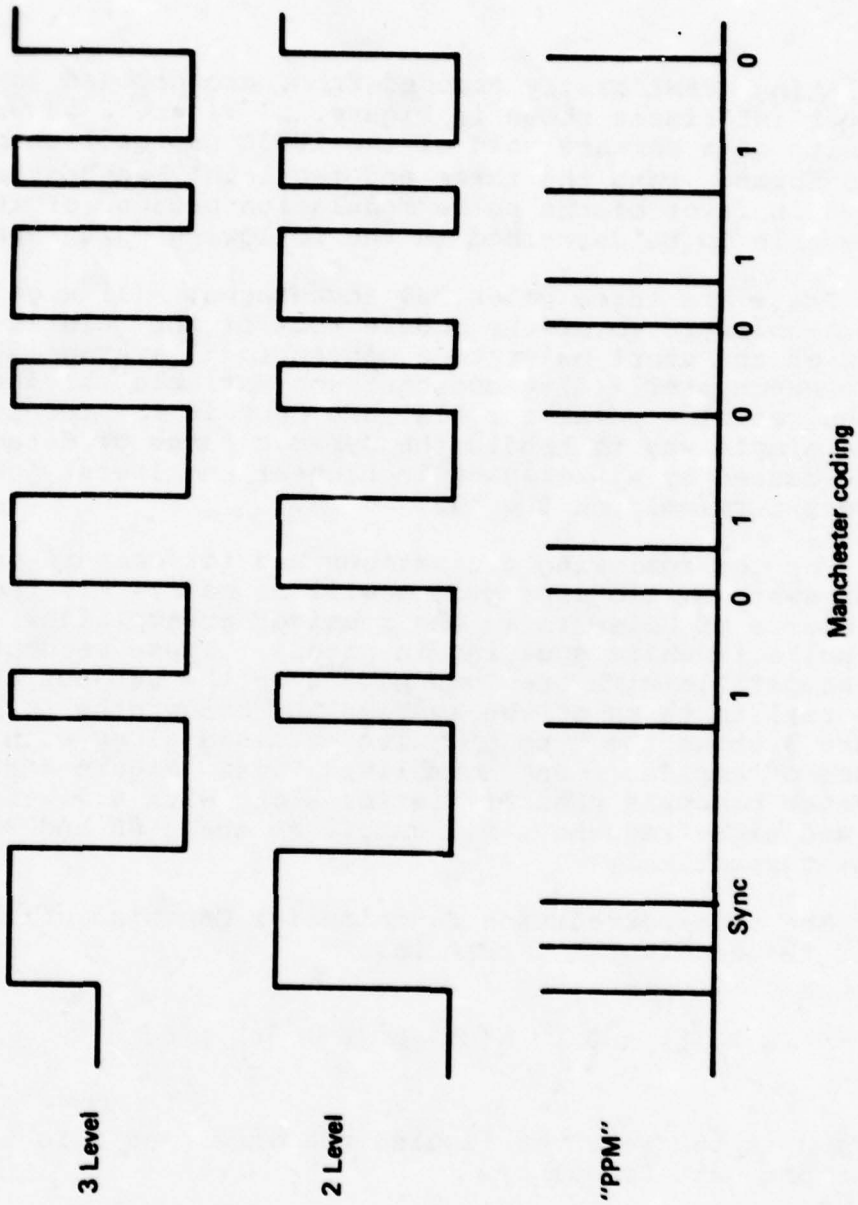



Figure 2. MIL-STD-1553A Protocol Command Word

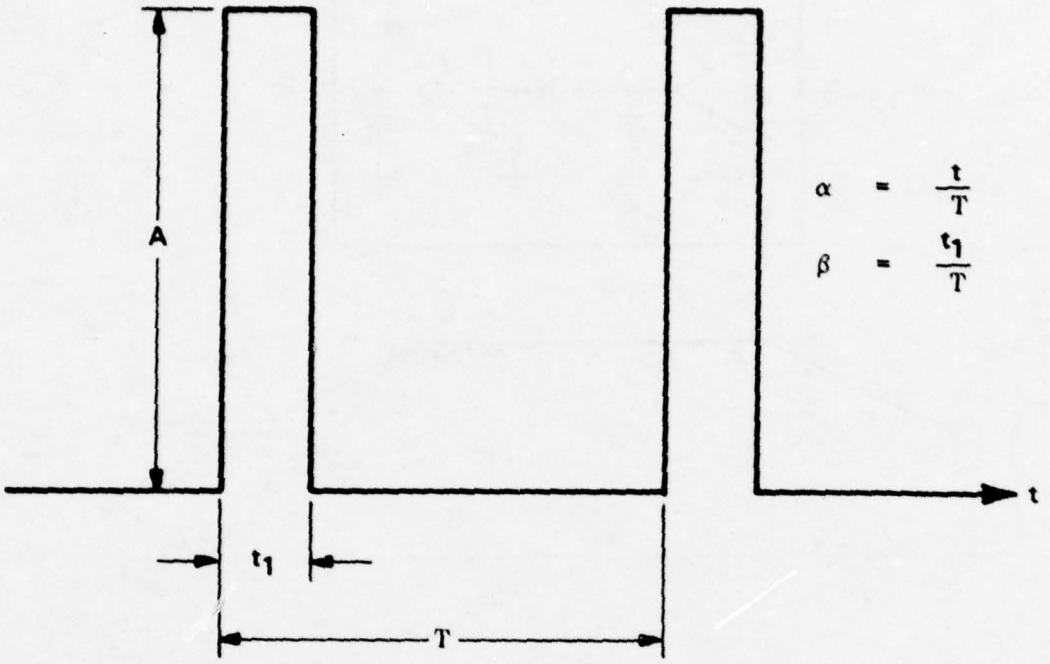



Figure 3. Amplitude as a Function of Normalized Time

A figure-of-merit (FOM) for this pulse system can be chosen as:

$$FOM = \frac{A'}{\sigma}$$

where A' is the pulse amplitude at the receiver's output.

Substituting for $\sigma$, and multiplying numerator and denominator by $A\sqrt{t_1}$

$$FOM = \frac{\sqrt{A} \cdot \sqrt{At_1}}{\sqrt{\mathcal{N}_0}} \cdot \frac{A'}{A\sqrt{t_1 \int_0^\infty [h(t)]^2 dt}}$$

where A is the input pulse amptitude $t_1$ is the pulse width.

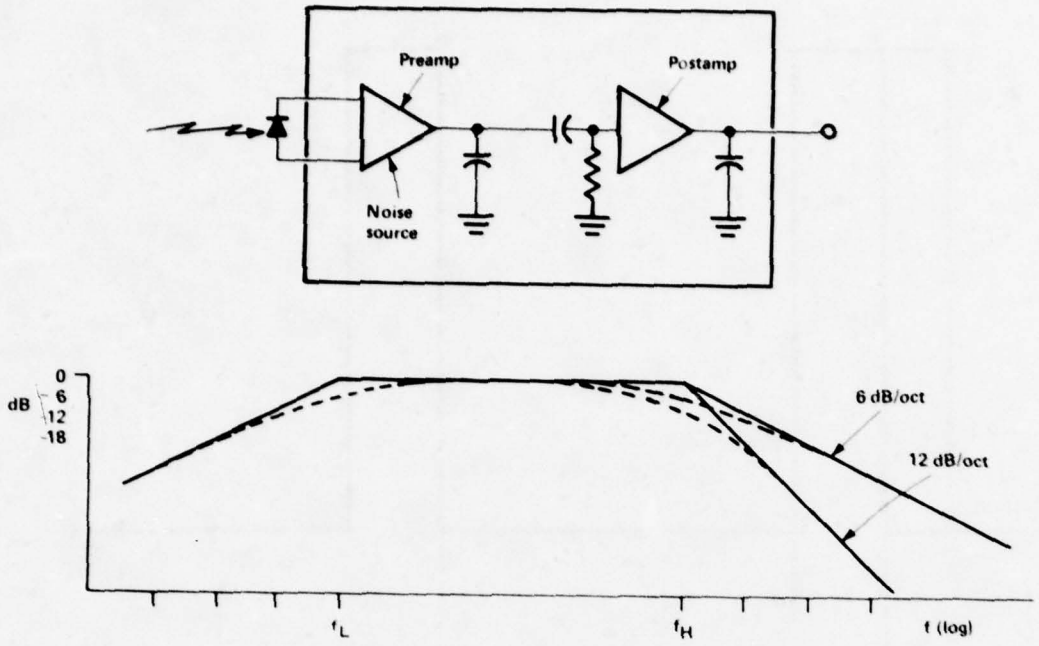



Figure 4. Receiver Bandpass Characteristics

To maintain a constant energy pulse to keep the temperature rise in the LED junction from degrading its performance or lifetime, the product $At_1$, must be kept constant. Since for a given receiver, $\emptyset$ is a constant, we can write the last equation as:

$$FOM = C \cdot \sqrt{A} \cdot A_n$$

where $A_n$ is a normalized output $\leq 1$ and $C = \sqrt{\frac{At_1}{\emptyset}}$

It follows, then, that the FOM is a function of the receiver bandpass characteristics, the pulse-width, and the input amplitude. The significant factor here is the gain in FOM by the square root of the input pulse amplitude. It also turns out that for a given pulse width and duty cycle, there is an optimum bandpass with a precisely located upper and lower cut-off frequency to give the maximum FOM.

When comparing a normal 50% average duty cycle Manchester with the PPM system using a 50-ns pulsewidth for a 1-MHz bit rate, the term $A_n$ for both systems when optimized for bandwidth is about the same ($\approx$0.85). However, with a 50-ns pulse, one can allow an increase of 10 times the amplitude for a factor of $\sqrt{10}$ or + 5 dB of peak power or peak signal-to-noise ratio. For LEDs, in their normal operating range, this requires a 1-A pulse 50 ns wide. Pulse-widths much less than 50 ns require very fast LEDs and correspondingly more difficult peak currents to generate. The pulse was therefore limited in this study to 50 ns and 1 A. Dr. J.R. Biard of Spectronics has performed an analysis in a paper "Thermal Transients in Light-Emitting Diodes," which shows that 1-A, 50-ns, 5% duty-cycle pulses maintain the same junction temperature rise as a 0.1 A 50% duty-cycle pulse. He maintains this will preserve conditions that will not degrade the life of the device.

A fourth advantage of PPM not mentioned before is the compatibility with a natural operating mode of lasers. When semiconductor laser technology has matured, the emitter can be easily replaced by a laser with minimum modifications to the system and gain additional dB through the $\sqrt{A}$ factor.

The equation for FOM involving constant $At_1$ and the $\sqrt{A}$ improvement verify conclusions reached in the literature that the narrower the pulse the better.(1)

A further advantage, as previously stated, is the fact that with the bandwidths involved, the receiver has essentially recovered between pulses such that a changing threshold for 1/0 decisions is not required because of the dynamic range as is required in the 50% duty cycle case. One sets the threshold for the minimum signal and leaves it there.

A third advantage related to the second is the method for handling the large dynamic signal range. To adjust the threshold levels in the 50% duty cycle case, a fast AGC system is required since the threshold must be set in the middle of the signal amplitude. In the case of the short pulse, the threshold is constant and all that need be done is to compress the high signal levels with techniques such as logarithmic amplifiers or networks.

As to bit error rate (BER), the following derivations for the PPM approach show the required signal-to-noise levels and the placement of the threshold at the 1/0 decision point.

In the PPM system we are concerned with two types of errors; the probability of a missed detection and the probability of a false alarm.

The differential false alarm probability, dp, that the noise exceeds some threshold, $k\sigma$ ($\sigma$-rms noise), between a time t and (t + dt) is:

$$dp = \varepsilon^{-\frac{1}{2}k^2} \bar{f}\, dt$$

where $\bar{f}$ is defined as the average frequency of receiver bandpass noise spectrum and is given by:

$$\bar{f} = \sqrt{\frac{\int_{-\infty}^{\infty} f^2 \phi_o(f)\, df}{\int_{-\infty}^{\infty} \phi_o(f)\, df}}$$

where $\phi_o(f)$ is the Fourier transform of the noise autocorrelation function $R_o(t)$ given before and is therefore the spectrum of noise in the filter output. The average number of threshold crossings, P, in time t is:

$$P = \int_o^t \bar{f}\, \varepsilon^{-\frac{1}{2}k^2}\, dt$$

If we normalize time as in Figure 3., and define a term $\bar{x} = 2\pi\bar{f}T$ where T is the reciprocal of the average pulse repitition frequency, then:

$$P = \int_o^{\alpha} \frac{\bar{x}}{2\pi}\, \varepsilon^{-\frac{1}{2}k^2}\, d\alpha$$

and $P = \frac{\bar{x}}{2\pi}\, \varepsilon^{-\frac{1}{2}k^2}$ for $\alpha = 1$

where:

$$\bar{x} = \sqrt{\frac{\int_o^{\infty} [h'(t)]^2 dt}{\int_o^{\infty} [h(t)]^2 dt}}$$

where h(t) is the impulse response of the receiver and h'(t) is its first derivative.

The equation for P describing the average number of threshold crossings in time t did not limit the duration of the threshold crossing. Since the 1/0 decision in reality is a function of the electronic response of the decision making circuit, the noise must remain some minimum time, $t_o$, above the threshold in order for a false alarm to occur. The true false alarm probability $P_{fa}$ is therefore:

$$P_{fa} = P' \cdot P = P' \cdot \frac{\bar{x}}{2\pi} \varepsilon^{-\frac{1}{2}k^2}$$

where P' is the conditional probability that if the noise exceeds the threshold $k\sigma$, it will stay above it for at least a time duration $t_o$. Reference (2) gives an equation for P' as:

$$P' \cong \varepsilon^{-\frac{1}{2}(\pi \bar{f} t_o k)^2} = \varepsilon^{-\frac{1}{2}(\bar{x} t_o k/2T)^2}$$

The false alarm probability is therefore:

$$P_{fa} \cong \frac{\bar{x}}{2\pi} \varepsilon^{-\frac{1}{2}k^2 [1 + (\bar{x} t_o/2T)^2]}$$

Or, for a given $P_{fa}$, the threshold, $k\sigma$, must be set at:

$$k\sigma = \sigma \sqrt{\frac{2(\ln\frac{\bar{x}}{2\pi} - \ln P_{fa})}{1 + \left(\frac{\bar{x}\, t_o}{2T}\right)^2}}$$

The probability of a missed detection, $P_{md}$, is:

$P_{md}$ = Probability that $[A' + n(t)] < k\sigma$ for some time t in the interval $t_o$. Since n(t) is the Gaussian variable with standard deviation $\sigma$, then:

$$P_{md} \leq \int_{\Delta}^{\infty} \varepsilon^{-\frac{1}{2}u^2} \, du$$

where $\Delta = \frac{1}{\sigma}(A' - k\sigma)$

This is the well known error function, erfc.

In a 1553B bus system using PPM, the average pulse spacing during a transmission is 1 μs. Under these conditions with a pulsewidth of 50 ns, the bandwidth, and upper and lower cutoff frequencies are 8.4, 8.8, and 0.4 MHz respectively. From the previous equations for $P_{fa}$ and $P_{md}$, if it is desired that the BER should be $10^{-12}$, then the threshold, $k\sigma$, must be $7.15\sigma$ for false alarms and the pulse amplitude, A', must be $7.03\sigma + 7.15\sigma = 14.18\sigma$. This is a limiting value since false alarms do not occur during pulses, and pulses have finite widths greater than the response times of the decision circuitry. It is additionally pessimistic since it does not take into account detection of errors by parity.

The threshold and pulse amplitudes for a 50% duty cycle straight Manchester are about the same. However, it should be remembered that the PPM pulse using LEDs can achieve $\sqrt{10}$ times the amplitude and therefore have an advantage of $\sqrt{10}$ or +5 dB in signal-to-noise ratio and +10 dB in terms of amplitude. This will add 10 dBm to the detector powers in Table 2 under system losses.

The transmitter and receiver circuitry are relatively simple and straightforward. Figure 5 is the schematic of the transmitter circuitry. Use of a VMOS power FET to switch a high current through the LED supplied from the energy stored in the capacitor C allows very fast risetimes. The resistor, $R_V$, is a negative temperature coefficient device which allows charging the capacitor C to higher voltages as the temperature rises, thereby offsetting the temperature characteristics of the VMOS and LED and tending to maintain a constant light output from the LED with temperature.

Figure 6 is a block diagram of the receiver circuit. The preamplifier is a transimpedance amplifier with a gain of 2.2 kΩ. The postamplifier is a differential amplifier with a gain of 10. The low-frequency cut-off is implemented at the postamplifier input with the RC coupling time constant. Two stages of amplification-compression follow the postamplifier. Figure 7 is a detailed schematic of one stage.

T1 and T2 together form a differential amplifier with a maximum pulse output of 4 V. T3 is an emitter follower driven by the differential amplifier, and at high levels it causes enough current through diode, $D_2$, to make it a low impedance compared to $R_2$ and $R_1$ such that the output never gets beyond about 0.8 V with up to 4 V signal at the base of T3.

With a 20-dB dynamic range in, this circuit will compress the range to around 11 dB. A second stage after amplification

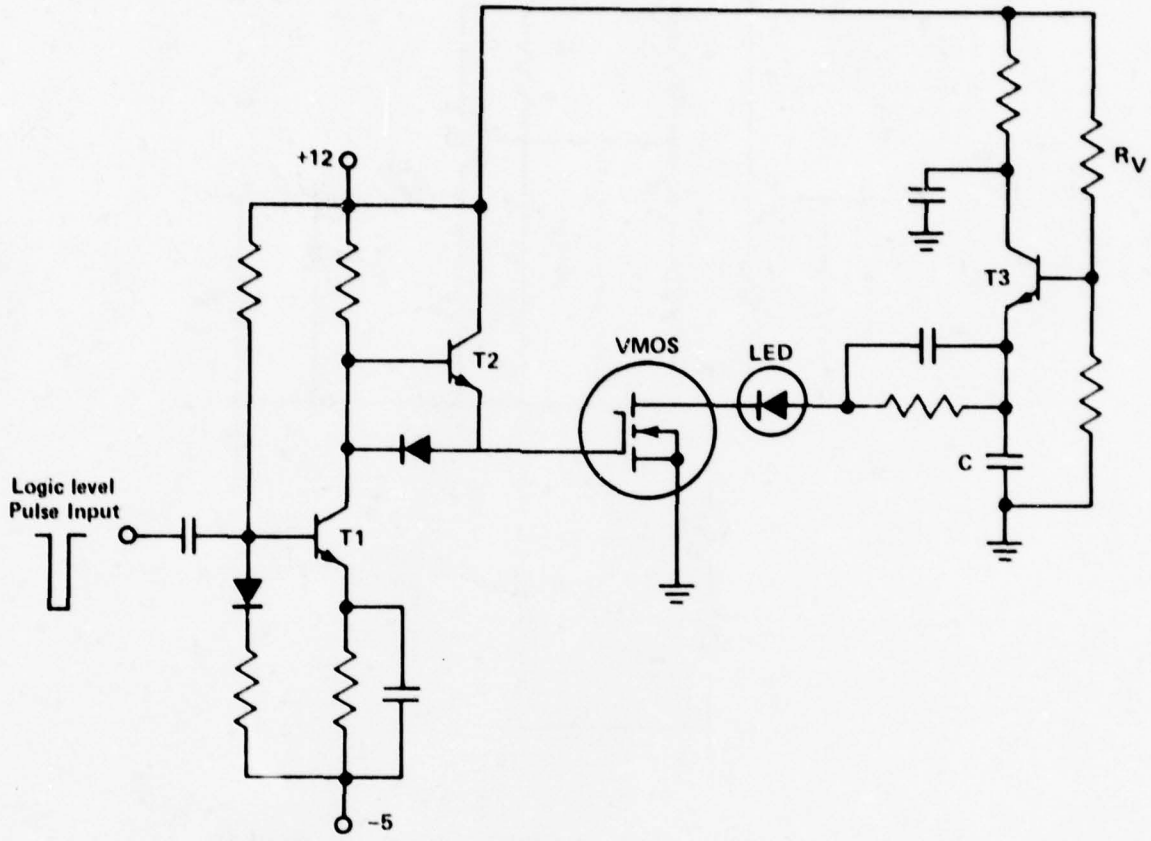



Figure 5. Transmitter Schematic

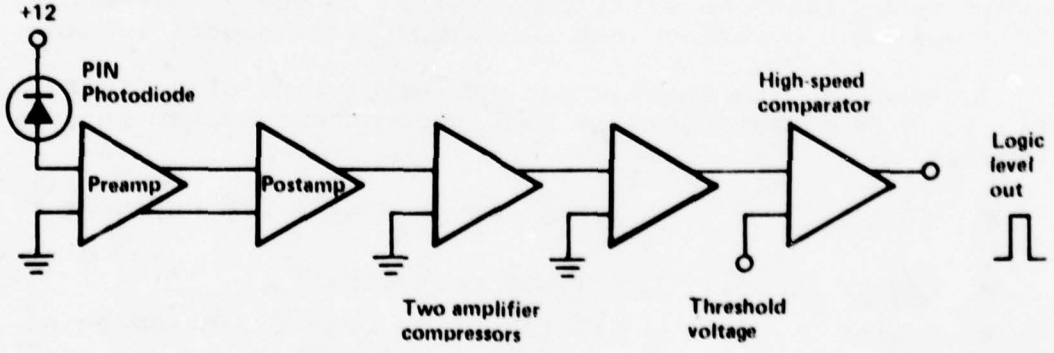



Figure 6. Receiver Block Diagram

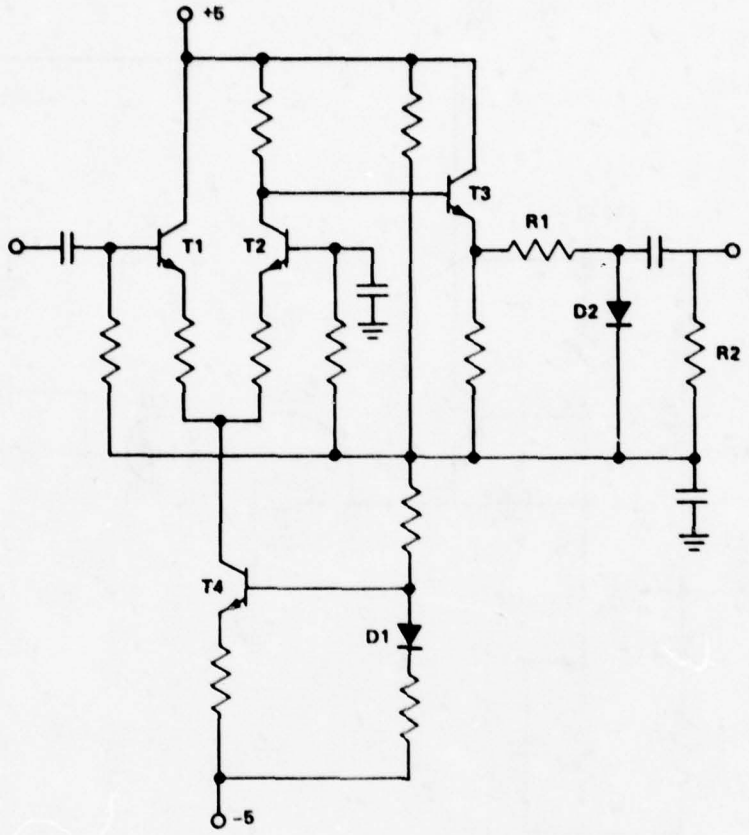



Figure 7. Amplifier - Compressor Schematic

of the 0.8 V to 4 V will further compress to a 4 or 5 dB dynamic range that can easily be handled in the comparator with a constant reference set for the minimum signal level.

In summary, the recommended implementation of a fiber optic 1553B - compatible data bus interconnect system is as follows:

- Source: Spectronics SE2231 LED or equivalent
- Fiber: 215-μm core PCS
- Cable: 19 fibers in ruggedized sheathing
- Connectors: Hexagonally packed stripped and thinly reclad fibers with keyed connection.
- Coupler: 16-port transmissive star

- Detector: Spectronics SE2232 PIN photodiode or equivalent
- Optical Waveform: PPM Manchester (a 50-ns pulse for every 500 ns the two-level Manchester is at an up level)
- Transmitter: A driver to furnish a 1-A pulse to the LED and driven by the PPM encoded data at TTL logic levels.
- Receiver: Transimpedance preamp, postamp, two stages of amplification and dynamic range compression, into a high-speed comparator with TTL logic levels out.
- Min., typical, Max., detector power: -33.1 dBm, -19.6 dBm, -9.9 dB.

This paper has described the effort in performance of a contract to IBM Federal Systems Division from the Naval Ocean System Center in San Diego, California.

## REFERENCES

(1) Receiver Design for Digital Fiber Optic Communication Systems, I by S.D. Personick BS&V Vol 52 No. 6. July-August 1973.

(2) Stationery and Stochastic Processes, Cramer & Leadbetter

MIL-STD-1553B
21 September 1978
SUPERSEDING
MIL-STD-1553A
30 April 1975

MILITARY STANDARD

# AIRCRAFT INTERNAL TIME DIVISION COMMAND/RESPONSE MULTIPLEX DATA BUS






FSC MISC

DEPARTMENT OF DEFENSE
Washington D.C. 20360

Aircraft Internal Time Division Command/Response Multiplex Data Bus

MIL-STD-1553B

1. This Military Standard is approved for use by all Department and Agencies of the Department of Defense.

2. beneficial comments (recommendations, additions, deletions) and any pertinent data which may be of use in improving this document should be addressed to: Aeronautical Systems Division, Attn: ENAl, Wright-Patterson Air Force base 45433, by using the self addressed Standardization Document Improvement Proposal (DD Form 1426) appearing at the end of this document or by letter.

## FOREWORD

This standard contains requirements for aircraft internal time division command/response multiplex data bus techniques which will be utilized in systems integration of aircraft subsystems. Even with the use of this standard, subtle differences will exist between multiplex data buses used on different aircraft due to particular aircraft mission requirements and the designer options allowed in this standard. The system designer must recognize this fact and design the multiplex bus controller hardware and software to accommodate such differences. These designer selected options must exist, so as to allow the necessary flexibility in the design of specific multiplex systems in order to provide for the control mechanism, architecture redundancy, degradation concept and traffic patterns peculiar to the specific aircraft mission requirements.

CONTENTS


| Paragraph | | Page |
|---|---|---|
| 1. | SCOPE | 1 |
| 1.1 | Scope | 1 |
| 1.2 | Application | 1 |
| 2. | REFERENCED DOCUMENTS | 1 |
| 2.1 | Issue of Document | 1 |
| 3. | DEFINITIONS | 1 |
| 3.1 | Bit | 1 |
| 3.2 | Bit Rate | 1 |
| 3.3 | Pulse Code Modulation (PCM) | 1 |
| 3.4 | Time Division Multiplexing (TDM) | 1 |
| 3.5 | Half Duplex | 1 |
| 3.6 | Word | 1 |
| 3.7 | Message | 3 |
| 3.8 | Subsystem | 3 |
| 3.9 | Data Bus | 3 |
| 3.10 | Terminal | 3 |
| 3.11 | Bus Controller | 3 |
| 3.12 | Bus Monitor | 3 |
| 3.13 | Remote Terminal (RT) | 3 |
| 3.14 | Asynchronous Operation | 3 |
| 3.15 | Dynamic Bus Control | 3 |
| 3.16 | Command/Response | 3 |
| 3.17 | Redundant Data Bus | 3 |
| 3.18 | Broadcast | 3 |
| 3.19 | Mode Code | 3 |
| 4. | GENERAL REQUIREMENTS | 4 |
| 4.1 | Test and Operating Requirements | 4 |
| 4.2 | Data Bus Operation | 4 |
| 4.3 | Characteristics | 4 |
| 4.3.1 | Data Form | 4 |
| 4.3.2 | Bit Priority | 4 |
| 4.3.3 | Transmission Method | 4 |
| 4.3.3.1 | Modulation | 4 |
| 4.3.3.2 | Data Code | 4 |
| 4.3.3.3 | Transmission Bit Rate | 4 |
| 4.3.3.4 | Word Size | 4 |
| 4.3.3.5 | Word Formats | 4 |
| 4.3.3.5.1 | Command Word | 8 |
| 4.3.3.5.1.1 | Sync | 8 |
| 4.3.3.5.1.2 | Remote Terminal Address | 8 |
| 4.3.3.5.1.3 | Transmit/Receive | 8 |
| 4.3.3.5.1.4 | Subaddress/Mode | 8 |
| 4.3.3.5.1.5 | Data word Count/Mode Code | 8 |
| 4.3.3.5.1.6 | Parity | 8 |
| 4.3.3.5.1.7 | Optional Mode Control | 8 |
| 4.3.3.5.1.7.1 | Dynamic Bus Control | 9 |

CONTENTS (Cont'd)


| Paragraph | | Page |
|---|---|---|
| 4.3.3.5.1.7.2 | Synchronize (Without Data Word) | 9 |
| 4.3.3.5.1.7.3 | Transmit Status Word | 9 |
| 4.3.3.5.1.7.4 | Initiate Self Test | 9 |
| 4.3.3.5.1.7.5 | Transmitter Shutdown | 9 |
| 4.3.3.5.1.7.6 | Override Transmitter Shutdown | 9 |
| 4.3.3.5.1.7.7 | Inhibit T/F Bit | 9 |
| 4.3.3.5.1.7.8 | Override Inhibit T/F Bit | 9 |
| 4.3.3.5.1.7.9 | Reset Remote Terminal | 9 |
| 4.3.3.5.1.7.10 | Reserved Mode Codes (01001 to 01111) | 9 |
| 4.3.3.5.1.7.11 | Transmit Vector Word | 11 |
| 4.3.3.5.1.7.12 | Synchronize (With Data Word) | 11 |
| 4.3.3.5.1.7.13 | Transmit Last Command Word | 11 |
| 4.3.3.5.1.7.14 | Transmit Built-In-Test (BIT) Word | 11 |
| 4.3.3.5.1.7.15 | Selected Transmitter Shutdown | 11 |
| 4.3.3.5.1.7.16 | Override Selected Transmitter Shutdown | 11 |
| 4.3.3.5.1.7.17 | Reserved mode Codes (10110 to 11111) | 11 |
| 4.3.3.5.2 | Data Word | 11 |
| 4.3.3.5.2.1 | Sync | 11 |
| 4.3.3.5.2.2 | Data | 12 |
| 4.3.3.5.2.3 | Parity | 12 |
| 4.3.3.5.3 | Status Word | 12 |
| 4.3.3.5.3.1 | Sync | 12 |
| 4.3.3.5.3.2 | RT Address | 12 |
| 4.3.3.5.3.3 | Message Error Bit | 12 |
| 4.3.3.5.3.4 | Instrumentation Bit | 12 |
| 4.3.3.5.3.5 | Service Request Bit | 12 |
| 4.3.3.5.3.6 | Reserved Status Bits | 12 |
| 4.3.3.5.3.7 | Broadcast Command Received Bit | 13 |
| 4.3.3.5.3.8 | Busy Bit | 13 |
| 4.3.3.5.3.9 | Subsystem Flag Bit | 13 |
| 4.3.3.5.3.10 | Dynamic Bus Control Acceptance Bit | 13 |
| 4.3.3.5.3.11 | Terminal Flag Bit | 13 |
| 4.3.3.5.3.12 | Parity Bit | 13 |
| 4.3.3.5.4 | Status Word Reset | 13 |
| 4.3.3.6 | Message Formats | 13 |
| 4.3.3.6.1 | Bus controller to Remote Terminal Transfers | 14 |
| 4.3.3.6.2 | Remote Terminal to Bus Controller Transfers | 14 |
| 4.3.3.6.3 | Remote Terminal to Remote Terminal Transfers | 14 |
| 4.3.3.6.4 | Mode Command Without Data Word | 14 |
| 4.3.3.6.5 | Mode Command With Data Word (Transmit) | 14 |
| 4.3.3.6.6 | Mode Command With Data Word (Receive) | 14 |
| 4.3.3.6.7 | Optional Broadcast Command | 14 |
| 4.3.3.6.7.1 | Bus controller to Remote Terminal(s) Transfer (Broadcast) | 14 |
| 4.3.3.6.7.2 | Remote Terminal to Remote Terminal(s) Transfer (Broadcast) | 17 |
| 4.3.3.6.7.3 | Mode Command Without Data Word (Broadcast) | 17 |
| 4.3.3.6.7.4 | Mode Command With Data Word (Broadcast) | 17 |
| 4.3.3.7 | Intermessage Gap | 17 |
| 4.3.3.8 | Response Time | 17 |

CONTENTS (Cont'd)


| Paragraph | | Page |
|---|---|---|
| 4.3.3.9 | Minimum No-Response Time-Out | 17 |
| 4.4 | Terminal Operation | 17 |
| 4.4.1 | Common Operation | 17 |
| 4.4.1.1 | Word Validation | 21 |
| 4.4.1.2 | Transmission Continuity | 21 |
| 4.4.1.3 | Terminal Fail-Safe | 21 |
| 4.4.2 | Bus Controller Operation | 21 |
| 4.4.3 | Remote Terminal | 21 |
| 4.4.3.1 | Operation | 21 |
| 4.4.3.2 | Superseding Valid Commands | 21 |
| 4.4.3.3 | Invalid Commands | 21 |
| 4.4.3.4 | Illegal Command | 21 |
| 4.4.3.5 | Valid Data Reception | 22 |
| 4.4.3.6 | Invalid Data Reception | 22 |
| 4.4.4 | Bus Monitor Operation | 22 |
| 4.5 | Hardware Characteristics | 22 |
| 4.5.1 | Data Bus Characteristics | 22 |
| 4.5.1.1 | Cable | 22 |
| 4.5.1.2 | Characteristics Impedance | 22 |
| 4.5.1.3 | Cable Attenuation | 22 |
| 4.5.1.4 | Cable Termination | 22 |
| 4.5.1.5 | Cable Stub Requirements | 22 |
| 4.5.1.5.1 | Transformer Coupled Stubs | 23 |
| 4.5.1.5.1.1 | Coupling Transformer | 23 |
| 4.5.1.5.1.1.1 | Transformer Input Impedance | 23 |
| 4.5.1.5.1.1.2 | Transformer Waveform Integrity | 23 |
| 4.5.1.5.1.1.3 | Transformer Common Mode Rejection | 23 |
| 4.5.1.5.1.2 | Fault Isolation | 23 |
| 4.5.1.5.1.3 | Cable Coupling | 23 |
| 4.5.1.5.1.4 | Stub Voltage Requirements | 23 |
| 4.5.1.5.2 | Direct Coupled Stubs | 23 |
| 4.5.1.5.2.1 | Fault Isolation | 23 |
| 4.5.1.5.2.2 | Cable Coupling | 25 |
| 4.5.1.5.2.3 | Stub Voltage Requirements | 25 |
| 4.5.1.5.3 | Wiring and Cabling for EMC | 25 |
| 4.5.2 | Terminal Characteristics | 25 |
| 4.5.2.1 | Terminals with Transformer Coupled Stubs | 25 |
| 4.5.2.1.1 | Terminal Output Characteristics | 25 |
| 4.5.2.1.1.1 | Output Levels | 25 |
| 4.5.2.1.1.2 | Output Waveform | 25 |
| 4.5.2.1.1.3 | Output Noise | 25 |
| 4.5.2.1.1.4 | Output Symmetry | 25 |
| 4.5.2.1.2 | Terminal Input Characteristics | 25 |
| 4.5.2.1.2.1 | Input Waveform Compatibility | 27 |
| 4.5.2.1.2.2 | Common Mode Rejection | 27 |
| 4.5.2.1.2.3 | Input Impedance | 27 |
| 4.5.2.1.2.4 | Noise Rejection | 27 |
| 4.5.2.2 | Terminals With Direct Coupled Stubs | 27 |
| 4.5.2.2.1 | Terminal Output Characteristics | 27 |
| 4.5.2.2.1.1 | Output Levels | 27 |
| 4.5.2.2.1.2 | Output Waveform | 29 |

CONTENTS (Cont'd)


| Paragraph | | Page |
|---|---|---|
| 4.5.2.2.1.3 | Output Noise | 29 |
| 4.5.2.2.1.4 | Output Symmetry | 29 |
| 4.5.2.2.2 | Terminal Input Characteristics | 29 |
| 4.5.2.2.2.1 | Input Waveform Compatibility | 29 |
| 4.5.2.2.2.2 | Common Mode Rejection | 29 |
| 4.5.2.2.2.3 | Input Impedance | 29 |
| 4.5.2.2.2.4 | Noise Rejection | 29 |
| 4.6 | Redundant Data Bus Requirements | 30 |
| 4.6.1 | Electrical Isolation | 30 |
| 4.6.2 | Single Event Failures | 30 |
| 4.6.3 | Dual Standby Redundant Data Bus | 30 |
| 4.6.3.1 | Data Bus Activity | 30 |
| 4.6.3.2 | Reset Data Bus Transmitter | 30 |
| 5. | DETAIL REQUIREMENTS | 30 |

| Paragraph | | Page |
|---|---|---|
| | FIGURES | |
| 1 | Sample Multiplex Data Bus Architecture | 2 |
| 2 | Data Encoding | 5 |
| 3 | Word Formats | 6 |
| 4 | Command and Status Sync | 7 |
| 5 | Data Sync | 7 |
| 6 | Information Transfer Formats | 15 |
| 7 | Broadcast Information Transfer Formats | 16 |
| 8 | Intermessage Gap and Response Time | 18 |
| 9 | Data Bus Interface Using Trans. Coupling | 19 |
| 10 | Data Bus Interface Using Direct Coupling | 20 |
| 11 | Coupling Transformer | 24 |
| 12 | Terminal I/O Characteristics for Transformer Coupled and Direct Coupled Stubs | 24 |
| 13 | Output Waveform | 26 |
| | TABLES | |
| I | Assigned Mode Codes | 10 |
| II | Criteria for Acceptance or Rejection of a Terminal for the Noise Rejection Test | 28 |
| | APPENDIX | |
| 10 | General | 31 |
| 10.1 | Redundancy | 31 |
| 10.2 | Bus Controller | 31 |
| 10.3 | Multiplex Selection Criteria | 33 |
| 10.4 | High Reliability Requirements | 33 |
| 10.5 | Stubbing | 33 |
| 10.6 | Use of Broadcast Option | 34 |
| | APPENDIX FIGURES | |
| 10.1 | Illustration of Possible Redundancy | 32 |
| 10.2 | Illustration of Possible Redundancy | 32 |

1. SCOPE

1.1 Scope. This standard establishes requirements for digital, command/response, time division multiplexing (Data Bus) techniques on aircraft. It encompasses the data bus line and its interface electronics illustrated on figure 1, and also defines the concept of operation and information flow on the multiplex data bus and the electrical and functional formats to be employed.

1.2 Application. When invoked in a specification or statement of work, these requirements shall apply to the multiplex data bus and associated equipment which is developed either alone or as a portion of an aircraft weapon system or subsystem development. The contractor is responsible for invoking all the applicable requirements of this Military Standard on any and all subcontractors he may employ.

2. REFERENCED DOCUMENTS

2.1 Issue of document. The following document, of the issue in effect on date of invitation for bid or request for proposal, forms a part of the standard to the extent specified herein.

SPECIFICATION

MILITARY

MIL-E-6051 Electromagnetic Compatibility Requirements, Systems

(Copies of specifications, standards, drawings, and publications required by contractors in connection with specific procurement functions should be obtained from the procuring activity or as directed by the contracting officer.)

3. DEFINITIONS

3.1 Bit. Contraction of binary digit: may be either zero or one. In information theory a binary digit is equal to one binary decision or the designation of one of two possible values or states of anything used to store or convey information.

3.2 Bit rate. The number of bits transmitted per second.

3.3 Pulse code modulation (PCM). The form of modulation in which the modulation signal is sampled, quantized, and coded so that each element of information consists of different types or numbers of pulses and spaces.

3.4 Time division multiplexing (TDM). The transmission of information from several signal sources through one communication system with different signal samples staggered in time to form a composite pulse train.

3.5 Half duplex. Operation of a data transfer system in either direction over a single line, but not in both directions on that line simultaneously.

3.6 Word. In this document a word is a sequence of 16 bits plus sync and parity. There are three types of words: command, status and data.

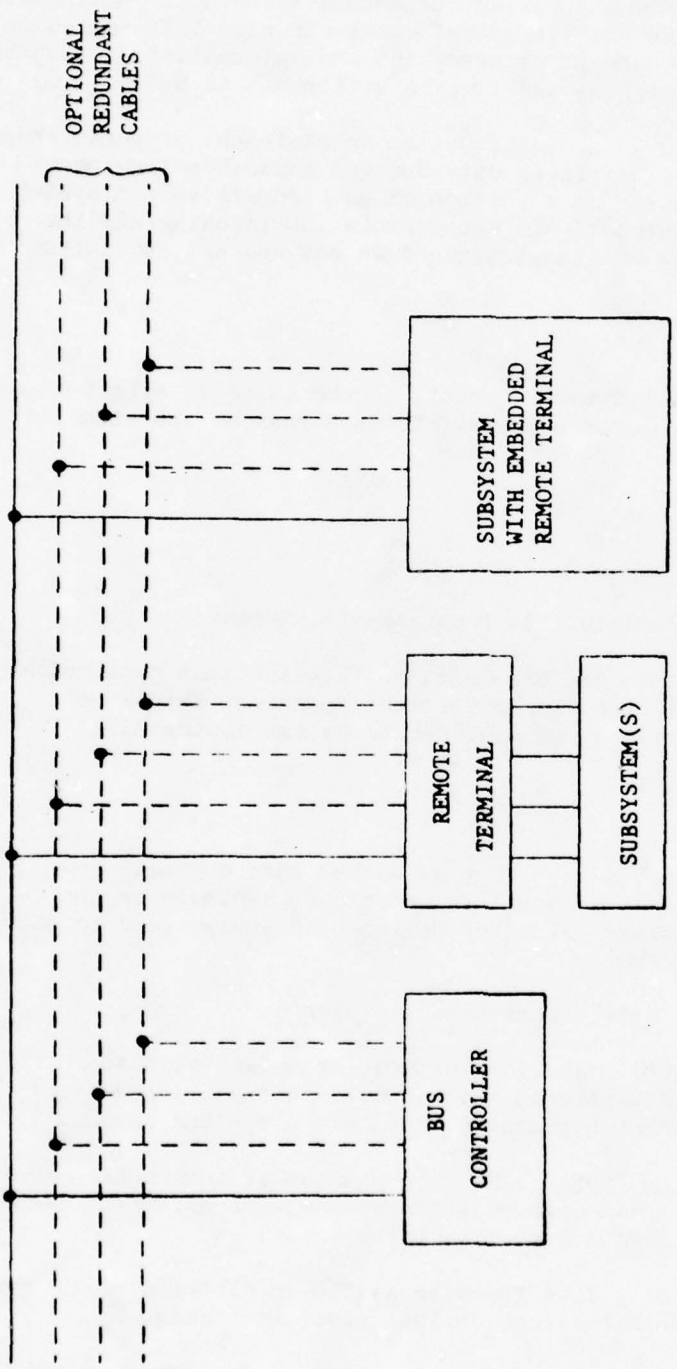



FIGURE 1. Sample multiplex data bus architecture.

3.7 Message. A single message is the transmission of a command word, status word, and data words if they are specified. For the case of a remote terminal to remote terminal (RT to RT) transmission, the message shall include the two command words, the two status words, and data words.

3.8 Subsystem. The device or functional unit receiving data transfer service from the data bus.

3.9 Data bus. Whenever a data bus or bus is referred to in this document it shall imply all the hardware including twisted shielded pair cables, isolation resistors, transformers, etc., required to provide a single data path between the bus controller and all the associated remote terminals.

3.10 Terminal. The electronic module necessary to interface the data bus with the subsystem and the subsystem with the data bus. Terminals may exist as separate line replaceable units (LRU's) or be contained within the elements of the subsystem.

3.11 Bus controller. The terminal assigned the task of initiating information transfers on the data bus.

3.12 Bus monitor. The terminal assigned the task of receiving bus traffic and extracting selected information to be used at a later time.

3.13 Remote terminal (RT). All terminals not operating as the bus controller or as a bus monitor.

3.14 Asynchronous operation. For the purpose of this standard, asynchronous operation is the use of an independent clock source in each terminal for message transmission. Decoding is achieved in receiving terminals using clock information derived from the message.

3.15 Dynamic bus control. The operation of a data bus system in which designated terminals are offered control of the data bus.

3.16 Command/Response. Operation of a data bus system such that remote terminals receive and transmit data only when commanded to do so by the bus controller.

3.17 Redundant data bus. The use of more than one data bus to provide more than one data path between the subsystems, i.e., dual redundant data bus, tri-redundant data bus, etc.

3.18 Broadcast. Operation of a data bus system such that information transmitted by the bus controller or a remote terminal is addressed to more than one of the remote terminals connected to the data bus.

3.19 Mode code. A means by which the bus controller can communicate with the multiplex bus related hardware, in order to assist in the management of information flow.

## 4. GENERAL REQUIREMENTS

4.1 Test and operating requirements. All requirements as specified herein shall be valid over the environmental conditions which the multiplex data bus system shall be required to operate.

4.2 Data bus operation. The multiplex data bus system in its most elemental configuration shall be as shown on figure 1. The multiplex data bus system shall function asynchronously in a command/response mode, and transmission shall occur in a half-duplex manner. Sole control of information transmission on the bus shall reside with the bus controller, which shall initiate all transmissions. The information flow on the data bus shall be comprised of messages which are, in turn, formed by three types of words (command, data, and status) as defined in 4.3.3.5.

4.3 Characteristics

4.3.1 Data form. Digital data may be transmitted in any desired form, provided that the chosen form shall be compatible with the message and word formats defined in this standard. Any unused bit positions in a word shall be transmitted as logic zeros.

4.3.2 Bit priority. The most significant bit shall be transmitted first with the less significant bits following in descending order of value in the data word. The number of bits required to define a quantity shall be consistent with the resolution or accuracy required. In the event that multiple precision quantities (information accuracy or resolution requiring more than 16 bits) are transmitted, the most significant bits shall be transmitted first, followed by the word(s) containing the lesser significant bits in numerical descending order. Bit packing of multiple quantities in a single data word is permitted.

4.3.3 Transmission method

4.3.3.1 Modulation. The signal shall be transferred over the data bus in serial digital pulse code modulation form.

4.3.3.2 Data code. The data code shall be Manchester II bi-phase level. A logic one shall be transmitted as a bipolar coded signal 1/0 (i.e., a positive pulse followed by a negative pulse). A logic zero shall be a bipolar coded signal 0/1 (i.e., a negative pulse followed by a positive pulse). A transition through zero occurs at the midpoint of each bit time (see figure 2).

4.3.3.3 Transmission bit rate. The transmission bit rate on the bus shall be 1.0 megabit per second with a combined accuracy and long-term stability of ± 0.1 percent (i.e., ± 1000 Hertz (Hz)). The short-term stability (i.e., stability over 1.0 second interval) shall be at least 0.01 percent (i.e., ± 100 Hz).

4.3.3.4 Word size. The word size shall be 16 bits plus the sync waveform and the parity bit for a total of 20 bits times as shown on figure 3.

4.3.3.5 Word formats. The word formats shall be as shown on figure 3 for the command, data, and status words.

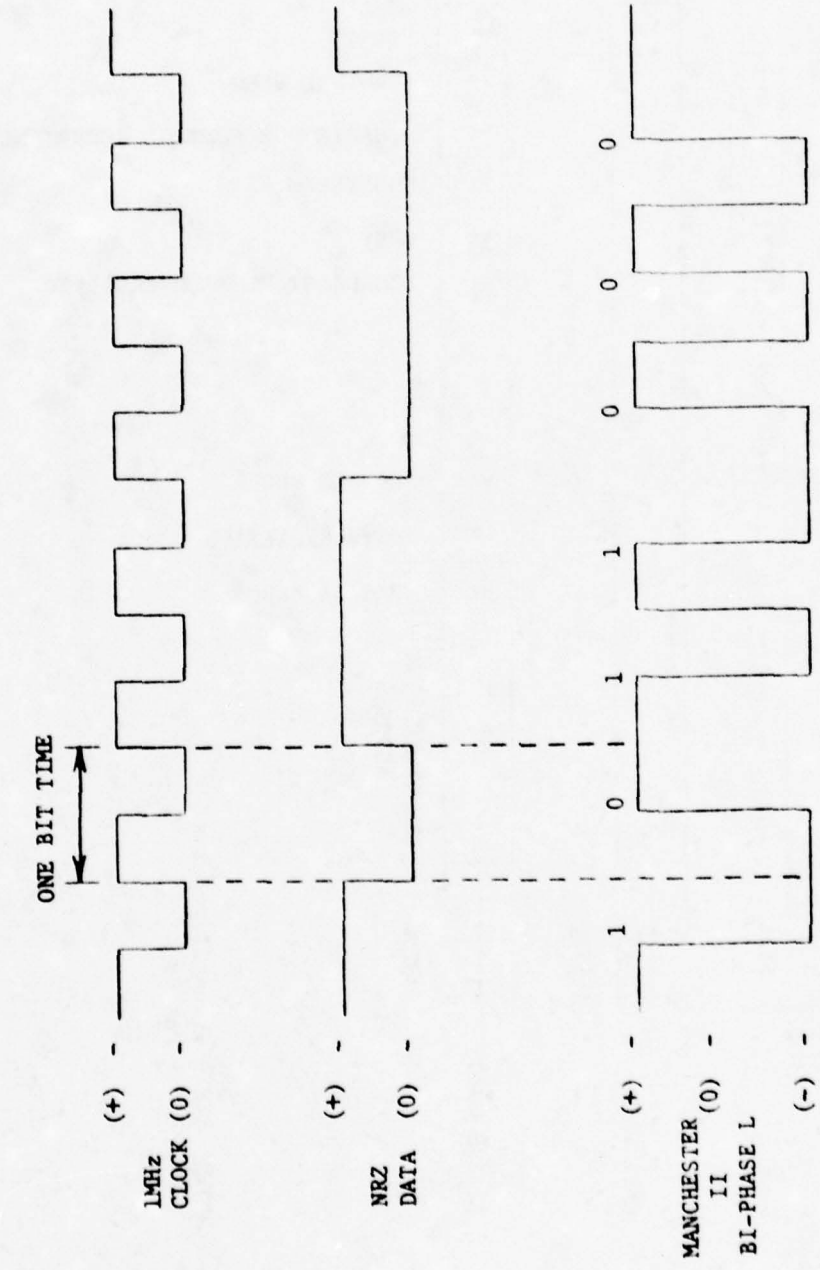
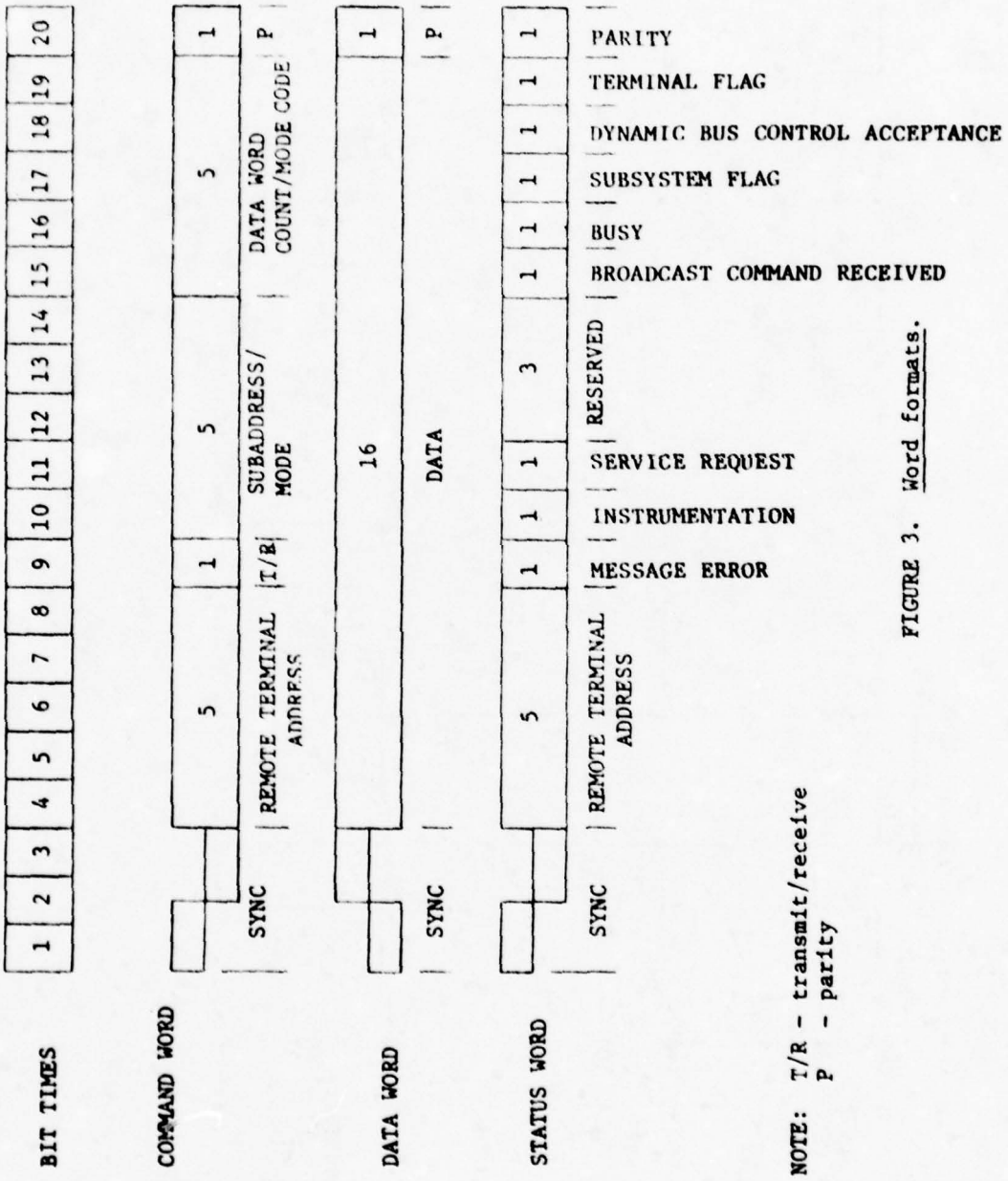
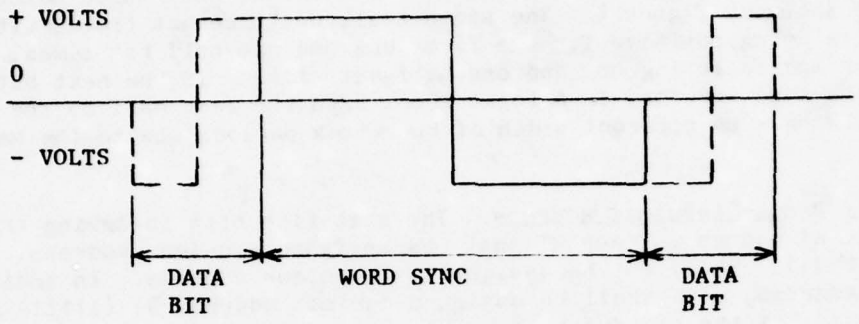



FIGURE 2. Data encoding.




NOTE: T/R - transmit/receive
P - parity

FIGURE 3. Word formats.




FIRURE 4. Command and status sync.

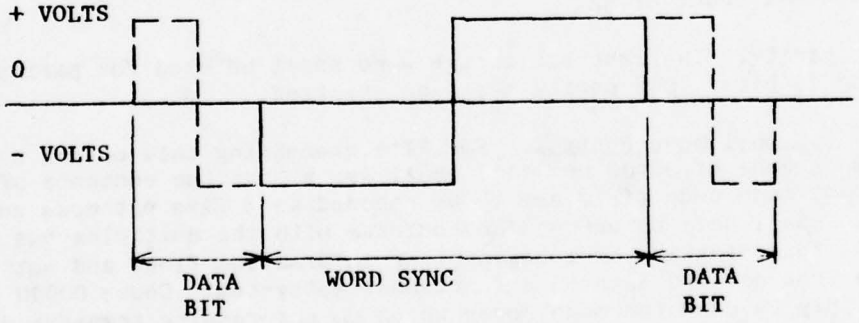



FIGURE 5. Data sync.

4.3.3.5.1 Command word. A command word shall be comprised of a sync waveform, remote terminal address field, transmit/receive (T/R) bit, subaddress/mode field, word count/mode code field, and a parity (P) bit (see figure 3).

4.3.3.5.1.1 Sync. The command sync waveform shall be an invalid Manchester waveform as shown on figure 4. The width shall be three bit times, with the sync waveform being positive for the first one and one-half bit times, and then negative for the following one and one-half bit times. If the next bit following the sync waveform is a logic zero, then the last half of the sync waveform will have an apparent width of two clock periods due to the Manchester encoding.

4.3.3.5.1.2 Remote terminal address. The next five bits following the sync shall be the RT address. Each RT shall be assigned a unique address. Decimal address 31 (11111) shall not be assigned as a unique address. In addition to its unique address, a RT shall be assigned decimal address 31 (11111) as the common address, if the broadcast option is used.

4.3.3.5.1.3 Transmit/receive. The next bit following the remote terminal address shall be the T/R bit, which shall indicate the action required of the RT. A logic zero shall indicate the RT is to receive, and a logic one shall indicate the RT is to transmit.

4.3.3.5.1.4 Subaddress/mode. The next five bits following the R/T bit shall be utilized to indicate an RT subaddress or use of mode control, as is dictated by the individual terminal requirements. The subaddress/mode values of 00000 and 11111 are reserved for special purposes, as specified in 4.3.3.5.1.7, and shall not be utilized for any other function.

4.3.3.5.1.5 Data word count/mode code. The next five bits following the subaddress/mode field shall be the quantity of data words to be either sent out or received by the RT or the optional mode code as specified in 4.3.3.5.1.7. A maximum of 32 data words may be transmitted or received in any one message block. All 1's shall indicate a decimal count of 31, and all 0's shall indicate a decimal count of 32.

4.3.3.5.1.6 Parity. The last bit in the word shall be used for parity over the preceding 16 bits. Odd parity shall be utilized.

4.3.3.5.1.7 Optional mode control. For RT's exercising this option a subaddress/mode code of 00000 or 11111 shall imply that the contents of the data word count/mode code field are to be decoded as a five bit mode command. The mode code shall only be used to communicate with the multiplex bus related hardware, and to assist in the management of information flow, and not to extract data from or feed data to a functional subsystem. Codes 00000 through 01111 shall only be used for mode codes which do not require transfer of a data word. For these codes, the T/R bit shall be set to 1. Codes 10000 through 11111 shall only be used for mode codes which require transfer of a single data word. For these mode codes, the T/R bit shall indicate the direction of data word flow as specified in 4.3.3.5.1.3. No multiple data word transfer shall be implemented with any mode code. The mode codes are reserved for the specific functions as specified in table I and shall not be used for any other purpose. If the designer chooses to implement any of these functions, the specific

codes, T/R bit assignments, and use of a data word, shall be used as indicated. The use of the broadcast command option shall only be applied to particular mode codes as specified in table I.

4.3.3.5.1.7.1 Dynamic bus control. The controller shall issue a transmit command to an RT capable of performing the bus control function. This RT shall respond with a status word as specified in 4.3.3.5.3. Control of the data bus passes from the offering bus controller to the accepting RT upon completion of the transmission of the status word by the RT. If the RT rejects control of the data bus, the offering bus controller retains control of the data bus.

4.3.3.5.1.7.2 Synchronize (without data word). This command shall cause the RT to synchronize (e.g., to reset the internal timer, to start a sequence, etc.). The RT shall transmit the status word as specified in 4.3.3.5.3.

4.3.3.5.1.7.3 Transmit status word. This command shall cause the RT to transmit the status word associated with the last valid command word preceding this command. This mode command shall not alter the state of the status word.

4.3.3.5.1.7.4 Initiate self test. This command shall be used to initiate self test within the RT. The RT shall transmit the status word as specified in 4.3.3.5.3.

4.3.3.5.1.7.5 Transmitter shutdown. This command (to only be used with dual redundant bus systems) shall cause the RT to disable the transmitter associated with the redundant bus. The RT shall not comply with a command to shut down a transmitter on the bus from which this command is received. In all cases, the RT shall respond with a status word as specified in 4.3.3.5.3 after this command.

4.3.3.5.1.7.6 Override transmitter shutdown. This command (to only be used with dual redundant bus system) shall cause the RT to enable a transmitter which was previously disabled. The RT shall not comply with a command to enable a transmitter on the bus from which this command is received. In all cases, the RT shall respond with a status word as specified in 4.3.3.5.3 after this command.

4.3.3.5.1.7.7 Inhibit terminal flag (T/F) bit. This command shall cause the RT to set the T/F bit in the status word specified in 4.3.3.5.3 to logic zero until otherwise commanded. The RT shall transmit the status word as specified in 4.3.3.5.3.

4.3.3.5.1.7.8 Override inhibit T/F bit. This command shall cause the RT to override the inhibit T/F bit specified in 4.3.3.5.1.7.7. The RT shall transmit the status word as specified in 4.3.3.5.3.

4.3.3.5.1.7.9 Reset remote terminal. This command shall be used to reset the RT to a power up initialized state. The RT shall first transmit its status word, and then reset.

4.3.3.5.1.7.10 Reserved mode codes (01001 to 01111). These mode codes are reserved for future use and shall not be used.

TABLE I. Assigned mode codes

| T/R Bit | Mode Code | Function | Associated Data Word | Broadcast Command Allowed |
|---|---|---|---|---|
| 1 | 00000 | Dynamic Bus Control | No | No |
| 1 | 00001 | Synchronize | No | Yes |
| 1 | 00010 | Transmit Status Word | No | No |
| 1 | 00011 | Initiate Self Test | No | Yes |
| 1 | 00100 | Transmitter Shutdown | No | Yes |
| 1 | 00101 | Override Transmitter Shutdown | No | Yes |
| 1 | 00110 | Inhibit Terminal Flag Bit | No | Yes |
| 1 | 00111 | Override Inhibit Terminal Flag bit | No | Yes |
| 1 | 01000 | Reset Remote Terminal | No | Yes |
| 1 | 01001 | Reserved | No | TBD |
| ↓ | ↓ | ↓ | ↓ | ↓ |
| 1 | 01111 | Reserved | No | TBD |
| 1 | 10000 | Transmit Vector Word | Yes | No |
| 0 | 10001 | Synchronize | Yes | Yes |
| 1 | 10010 | Transmit Last Command | Yes | No |
| 1 | 10011 | Transmit BIT Word | Yes | No |
| 0 | 10100 | Selected Transmitter Shutdown | Yes | Yes |
| 0 | 10101 | Override Selected Transmitter Shutdown | Yes | Yes |
| 1 or 0 | 10110 | Reserved | Yes | TBD |
| ↓ | ↓ | ↓ | ↓ | ↓ |
| 1 or 0 | 11111 | Reserved | Yes | TBD |

NOTE: To be determined (TBD)

4.3.3.5.1.7.11 Transmit vector word. This command shall cause the RT to transmit a status word as specified in 4.3.3.5.3 and a data word containing service request information.

4.3.3.5.1.7.12 Synchronize (with data word). The RT shall receive a command word followed by a data word as specified in 4.3.3.5.2. The data word shall contain synchronization information for the RT. After receiving the command and data word, the RT shall transmit the status word as specified in 4.3.3.5.3.

4.3.3.5.1.7.13 Transmit last command word. This command shall cause the RT to transmit its status word as specified in 4.3.3.5.3 followed by a single data word which contains bits 4-19 of the last command word, excluding a transmit last command word mode code received by the RT. This mode command shall not alter the state of the RT's status word.

4.3.3.5.1.7.14 Transmit built-in-test (BIT) word. This command shall cause the RT to transmit its status word as specified in 4.3.3.5.3 followed by a single data word containing the RT BIT data. This function is intended to supplement the available bits in the status word when the RT hardware is sufficiently complex to warrant its use. The data word, containing the RT BIT data, shall not be altered by the reception of a transmit last command or a transmit status word mode code. This function shall not be used to convey BIT data from the associated subsystem(s).

4.3.3.5.1.7.15 Selected transmitter shutdown. This command shall cause the RT to disable the transmitter associated with a specified redundant data bus. The command is designed for use with systems employing more than two redundant buses. The transmitter that is to be disabled shall be identified in the data word following the command word in the format as specified in 4.3.3.5.2. The RT shall not comply with a command to shut down a transmitter on the bus from which this command is received. In all cases, the RT shall respond with a status word as specified in 4.3.3.5.3.

4.3.3.5.1.7.16 Override selected transmitter shutdown. This command shall cause the RT to enable a transmitter which was previously disabled. The command is designed for use with systems employing more than two redundant buses. The transmitter that is to be enabled shall be identified in the data word following the command word in the format as specified in 4.3.3.5.2. The RT shall not comply with a command to enable a transmitter on the bus from which this command is received. In all cases, the RT shall respond with a status word as specified in 4.3.3.5.3.

4.3.3.5.1.7.17 Reserved mode codes (10110 to 11111). These mode codes are reserved for future use and shall not be used.

4.3.3.5.2 Data word. A data word shall be comprised of a sync waveform, data bits, and a parity bit (see figure 3).

4.3.3.5.2.1 Sync. The data sync waveform shall be an invalid Manchester waveform as shown on figure 5. The width shall be three bit times, with the waveform being negative for the first one and one-half bit times, and then positive for the following one and one-half bit times. Note that if the bits preceding and following the sync are logic ones, then the apparent width of the sync waveform will be increased to four bit times.

4.3.3.5.2.2 Data. The sixteen bits following the sync shall be utilized for data transmission as specified in 4.3.2.

4.3.3.5.2.3 Parity. The last bit shall be utilized for parity as specified in 4.3.3.5.1.6.

4.3.3.5.3 Status word. A status word shall be comprised of a sync waveform, RT address, message error bit, instrumentation bit, service request bit, three reserved bits, broadcast command received bit, busy bit, subsystem flag bit, dynamic bus control acceptance bit, terminal flag bit, and a parity bit. For optional broadcast operation, transmission of the status word shall be suppressed as specified in 4.3.3.6.7.

4.3.3.5.3.1 Sync. The status sync waveform shall be as specified in 4.3.3.5.1.1.

4.3.3.5.3.2 RT address. The next five bits following the sync shall contain the address of the RT which is transmitting the status word as defined in 4.3.3.5.1.2.

4.3.3.5.3.3 Message error bit. The status word bit at bit time nine (see figure 3) shall be utilized to indicate that one or more of the data words associated with the preceding receive command word from the bus controller has failed to pass the RT's validity tests as specified in 4.4.1.1. This bit shall also be set under the conditions specified in 4.4.1.2, 4.4.3.4 and 4.4.3.6. A logic one shall indicate the presence of a message error, and a logic zero shall show its absence. All RT's shall implement the message error bit.

4.3.3.5.3.4 Instrumentation bit. The status word at bit time ten (see figure 3) shall be reserved for the instrumentation bit and shall always be a logic zero. This bit is intended to be used in conjunction with a logic one in bit time ten of the command word to distinguish between a command word and a status word. The use of the instrumentation bit is optional.

4.3.3.5.3.5 Service request bit. The status word bit at bit time eleven (see figure 3) shall be reserved for the service request bit. The use of this bit is optional. This bit when used, shall indicate the need for the bus controller to take specific predefined actions relative to either the RT or associated subsystem. Multiple subsystems, interfaced to a single RT, which individually require a service request signal shall logically OR their individual signals into the single status word bit. In the event this logical OR is performed, then the designer must make provisions in a separate data word to identify the specific requesting subsystem. The service request bit is intended to be used only to trigger data transfer operations which take place on an exception rather than periodic basis. A logic one shall indicate the presence of a service request, and a logic zero its absence. If this function is not implemented, the bit shall be set to zero.

4.3.3.5.3.6 Reserved status bits. The status word bits at bit times twelve through fourteen are reserved for future use and shall not be used. These bits shall be set to a logic zero.

4.3.3.5.3.7 Broadcast command received bit. The status word at bit time fifteen shall be set to a logic one to indicate that the preceding valid command word was a broadcast command and a logic zero shall show it was not a broadcast command. If the broadcast command option is not used, this bit shall be set to a logic zero.

4.3.3.5.3.8 Busy bit. The status word bit at bit time sixteen (see figure 3) shall be reserved for the busy bit. The use of this bit is optional. This bit, when used, shall indicate that the RT or subsystem is unable to move data to or from the subsystem in compliance with the bus controller's command. A logic one shall indicate the presence of a busy condition, and a logic zero its absence. In the event the busy bit is set in response to a transmit command, then the RT shall transmit its status word only. If this function is not implemented, the bit shall be set to logic zero.

4.3.3.5.3.9 Subsystem flag bit. The status word bit at bit time seventeen (see figure 3) shall be reserved for the subsystem flag bit. The use of this bit is optional. This bit, when used, shall flag a subsystem fault condition, and alert the bus controller to potentially invalid data. Multiple subsystems, interfaced to a single RT, which individually require a subsystem flag bit signal shall logically OR their individual signals into the single status word bit. In the event this logical OR is performed, then the designer must make provisions in a separate data word to identify the specific reporting subsystem. A logic one shall indicate the presence of the flag, and a logic zero its absence. If not used, this bit shall be set to logic zero.

4.3.3.5.3.10 Dynamic bus control acceptance bit. The status word bit at bit time eighteen (see figure 3) shall be reserved for the acceptance of dynamic bus control. This bit shall be used if the RT implements the optional dynamic bus control function. This bit, when used, shall indicate acceptance or rejection of a dynamic bus control offer as specified in 4.3.3.5.1.7.1. A logic one shall indicate acceptance of control, and a logic zero shall indicate rejection of control. If this function is not used, this bit shall be set to logic zero.

4.3.3.5.3.11 Terminal flag bit. The status word bit at bit time nineteen (see figure 3) shall be reserved for the terminal flag function. The use of this bit is optional. This bit, when used, shall flag a RT fault condition. A logic one shall indicate the presence of the flag, and a logic zero, its absence. If not used, this bit shall be set to logic zero.

4.3.3.5.3.12 Parity bit. The least significant bit in the status word shall be utilized for parity as specified in 4.3.3.5.1.6.

4.3.3.5.4 Status word reset. The status word bit, with the exception of the address, shall be set to logic zero after a valid command word is received by the RT with the exception as specified in 4.3.3.5.1.7. If the conditions which caused bits in the status word to be set (e.g., terminal flag) continue after the bits are reset to logic zero, then the affected status word bit shall be again set, and then transmitted on the bus as required.

4.3.3.6 Message formats. The messages transmitted on the data bus shall be in accordance with the formats on figure 6 and figure 7. The maximum and minimum response times shall be as stated in 4.3.3.7 and 4.3.3.8. No message formats, other than those defined herein, shall be used on the bus.

4.3.3.6.1 Bus controller to remote terminal transfers. The bus controller shall issue a receive command followed by the specified number of data words. The RT shall, after message validation, transmit a status word back to the controller. The command and data words shall be transmitted in a contiguous fashion with no interword gaps.

4.3.3.6.2 Remote terminal to bus controller transfers. The bus controller shall issue a transmit command to the RT. The RT shall, after command word validation, transmit a status word back to the bus controller, followed by the specified number of data words. The status and data words shall be transmitted in a contiguous fashion with no interword gaps.

4.3.3.6.3 Remote terminal to remote terminal transfers. The bus controller shall issue a receive command to RT A followed contiguously by a transmit command to RT B. RT B shall, after command validation, transmit a status word followed by the specified number of data words. The status and data words shall be transmitted in a contiguous fashion with no gap. At the conclusion of the data transmission by RT B, RT A shall transmit a status word within the specified time period.

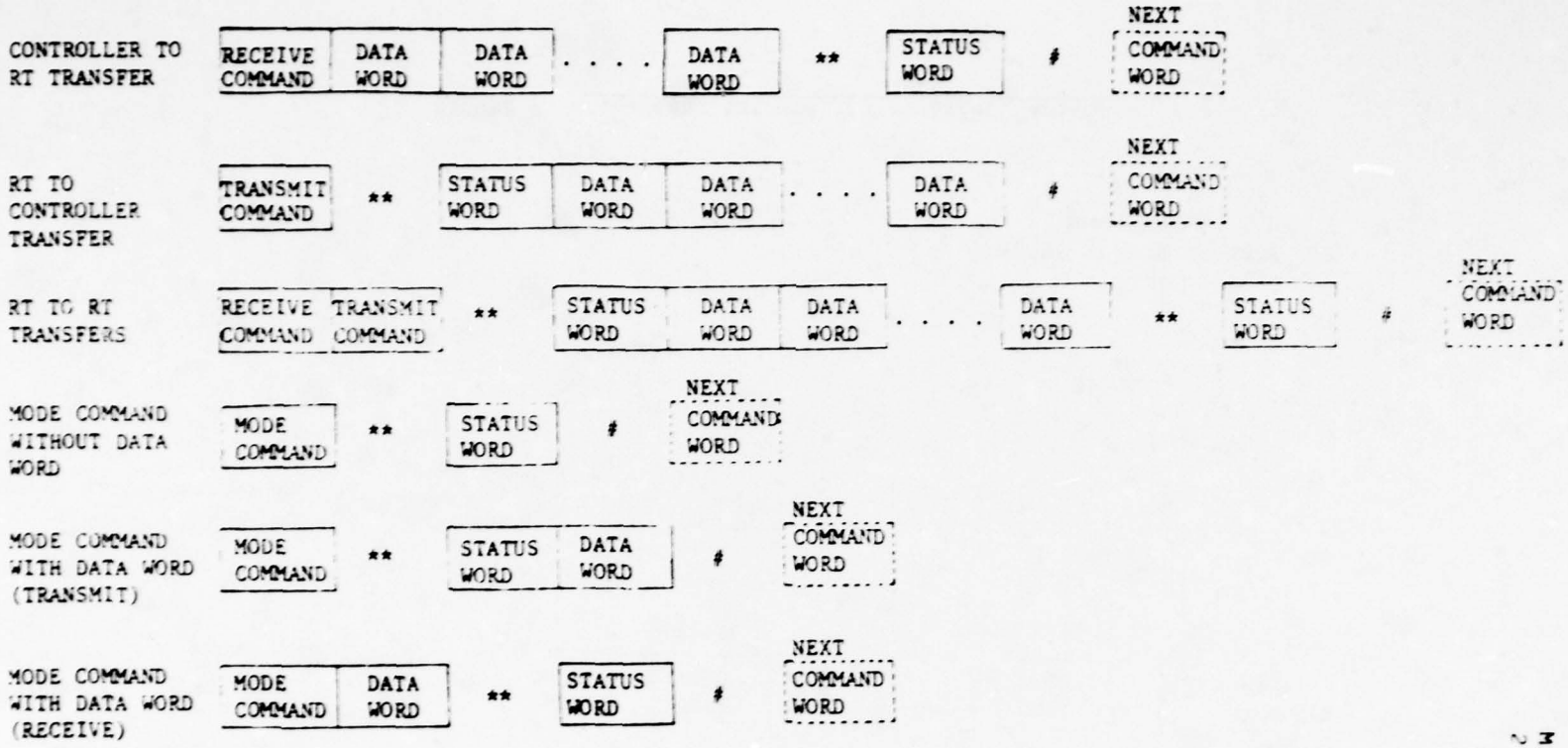
4.3.3.6.4 Mode command without data word. The bus controller shall issue a transmit command to the RT using a mode code specified in table I. The RT shall, after command word validation, transmit a status word.

4.3.3.6.5 Mode command with data word (transmit). The bus controller shall issue a transmit command to the RT using a mode code specified in table I. The RT shall, after command word validation, transmit a status word followed by one data word. The status word and data word shall be transmitted in a contiguous fashion with no gap.

4.3.3.6.6 Mode command with data word (receive). The bus controller shall issue a receive command to the RT using a mode code specified in table I, followed by one data word. The command word and data word shall be transmitted in a contiguous fashion with no gap. The RT shall, after command and data word validation, transmit a status word back to the controller.

4.3.3.6.7 Optional broadcast command. See 10.6 for additional information on the use of the broadcast command.

4.3.3.6.7.1 Bus controller to remote terminal(s) transfer (broadcast). The bus controller shall issue a receive command word with 11111 in the RT address field followed by the specified number of data words. The command word and data words shall be transmitted in a contiguous fashion with no gap. The RT(s) with the broadcast option shall after message validation, set the broadcast command received bit in the status word as specified in 4.3.3.5.3.7 and shall not transmit the status word.




NOTE: # INTERMESSAGE GAP
** RESPONSE TIME

FIGURE 6. Information transfer formats.

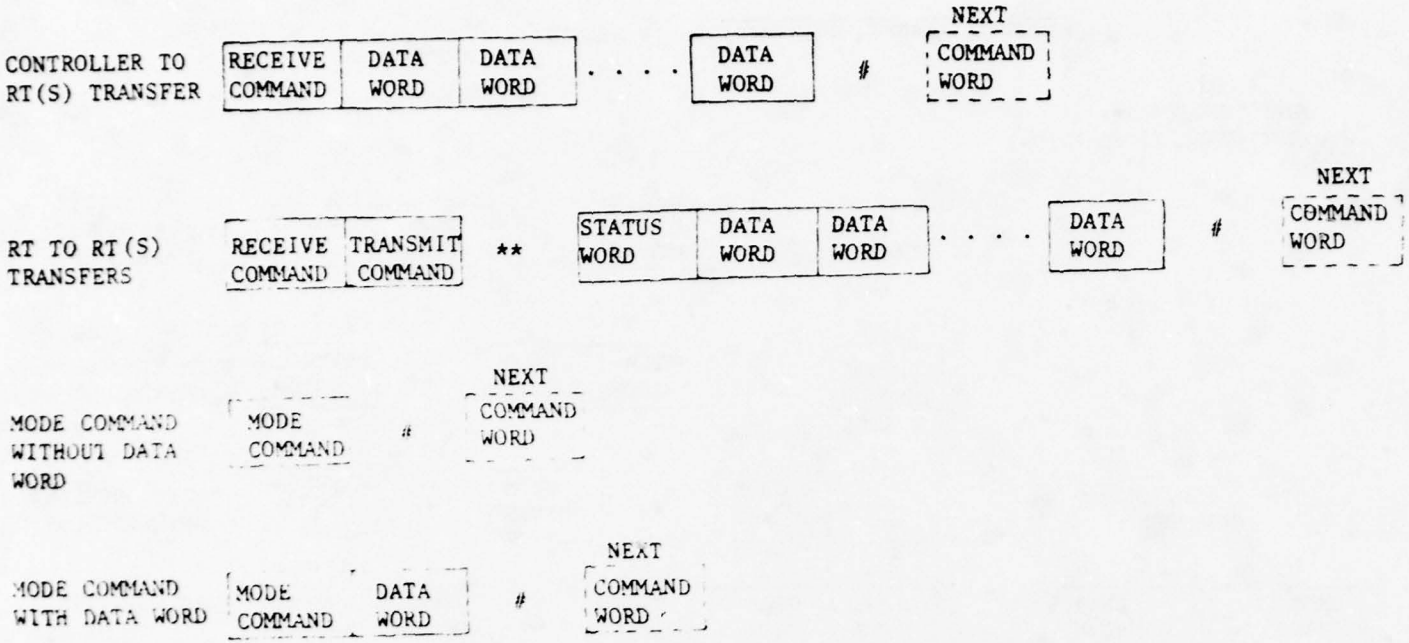



FIGURE 7. Broadcast information transfer formats.

4.3.3.6.7.2 Remote terminal to remote terminal(s) transfers (broadcast). The bus controller shall issue a receive command word with 11111 in the RT address field followed by a transmit command to RT A using the RT's address. RT A shall, after command word validation, transmit a status word followed by the specified number of data words. The status and data words shall be transmitted in a contiguous fashion with no gap. The RT(s) with the broadcast option, excluding RT A, shall after message validation, set the broadcast received bit in the status word as specified in 4.3.3.5.3.7 and shall not transmit the status word.

4.3.3.6.7.3 Mode command without data word (broadcast). The bus controller shall issue a transmit command word with 11111 in the RT address field, and a mode code specified in table I. The RT(s) with the broadcast option shall after command word validation, set the broadcast received bit in the status word as specified in 4.3.3.5.3.7 and shall not transmit the status word.

4.3.3.6.7.4 Mode command with data word (broadcast). The bus controller shall issue a receive command word with 11111 in the RT address field and a mode code specified in table I, followed by one data word. The command word and data word shall be transmitted in a contiguous fashion with no gap. The RT(s) with the broadcast option shall after message validation, set the broadcast received bit in the status word as specified in 4.3.3.5.3.7 and shall not transmit the status word.

4.3.3.7 Intermessage gap. The bus controller shall provide a minimum gap time of 4.0 microseconds (μs) between messages as shown on figure 6 and figure 7. This time period, shown as T on figure 8, is measured at point A of the bus controller as shown on figure 9 or figure 10. The time is measured from the mid-bit zero crossing of the last bit of the preceding message to mid-zero crossing of the next command word sync.

4.3.3.8 Response time. The RT shall respond, in accordance with 4.3.3.6, to a valid command word within the time period of 4.0 to 12.0 μs. This time period, shown as T on figure 8, is measured at point A of the RT as shown on figure 9 or figure 10. The time is measured from the mid bit zero crossing of the last word as specified in 4.3.3.6 and as shown on figure 6 and figure 7 to the mid-zero crossing of the status word sync.

4.3.3.9 Minimum no-response time-out. The minimum time that a terminal shall wait before considering that a response as specified in 4.3.3.8 has not occurred shall be 14.0 μs. The time is measured from the mid-bit zero crossing of the last bit of the last word to the mid-zero crossing of the expected status word sync at point A of the terminal as shown on figure 9 or figure 10.

4.4 Terminal operation.

4.4.1 Common operation. Terminals shall have common operating capabilities as specified in the following paragraphs.

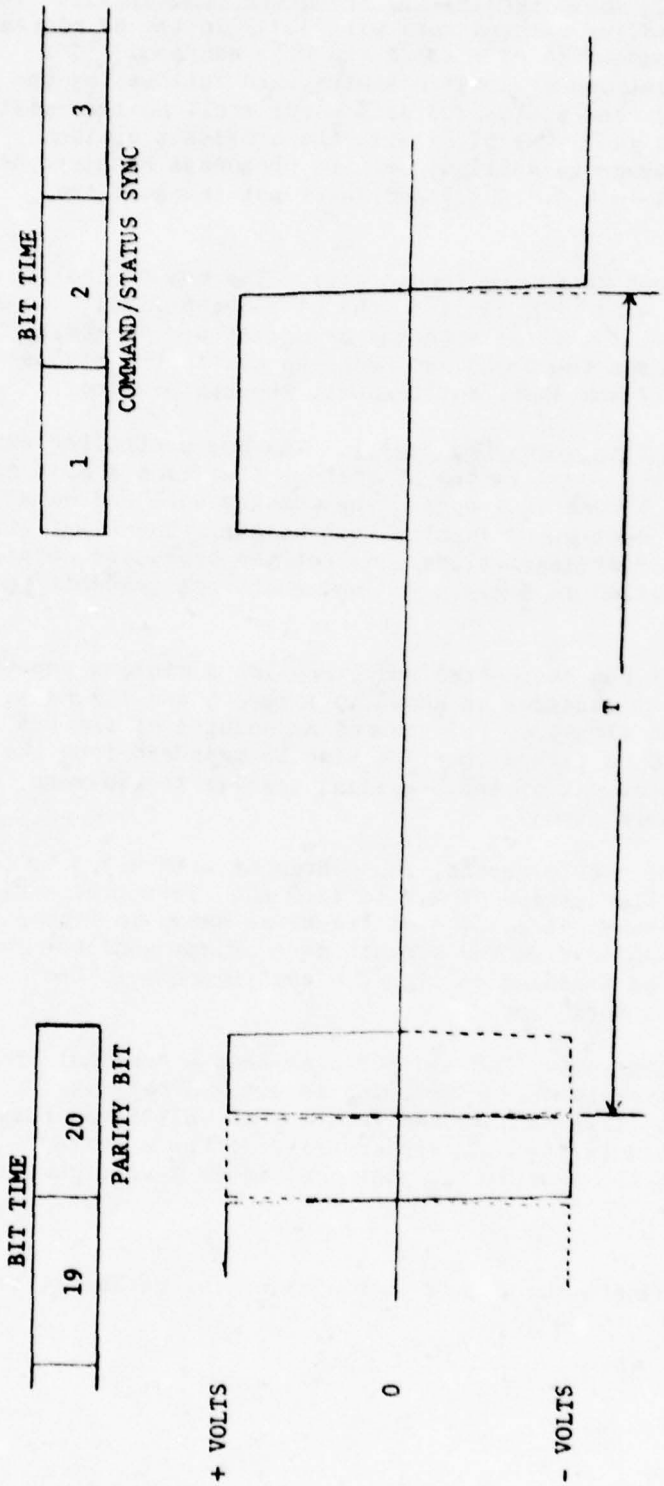



FIGURE 8. Intermessage gap and response time.

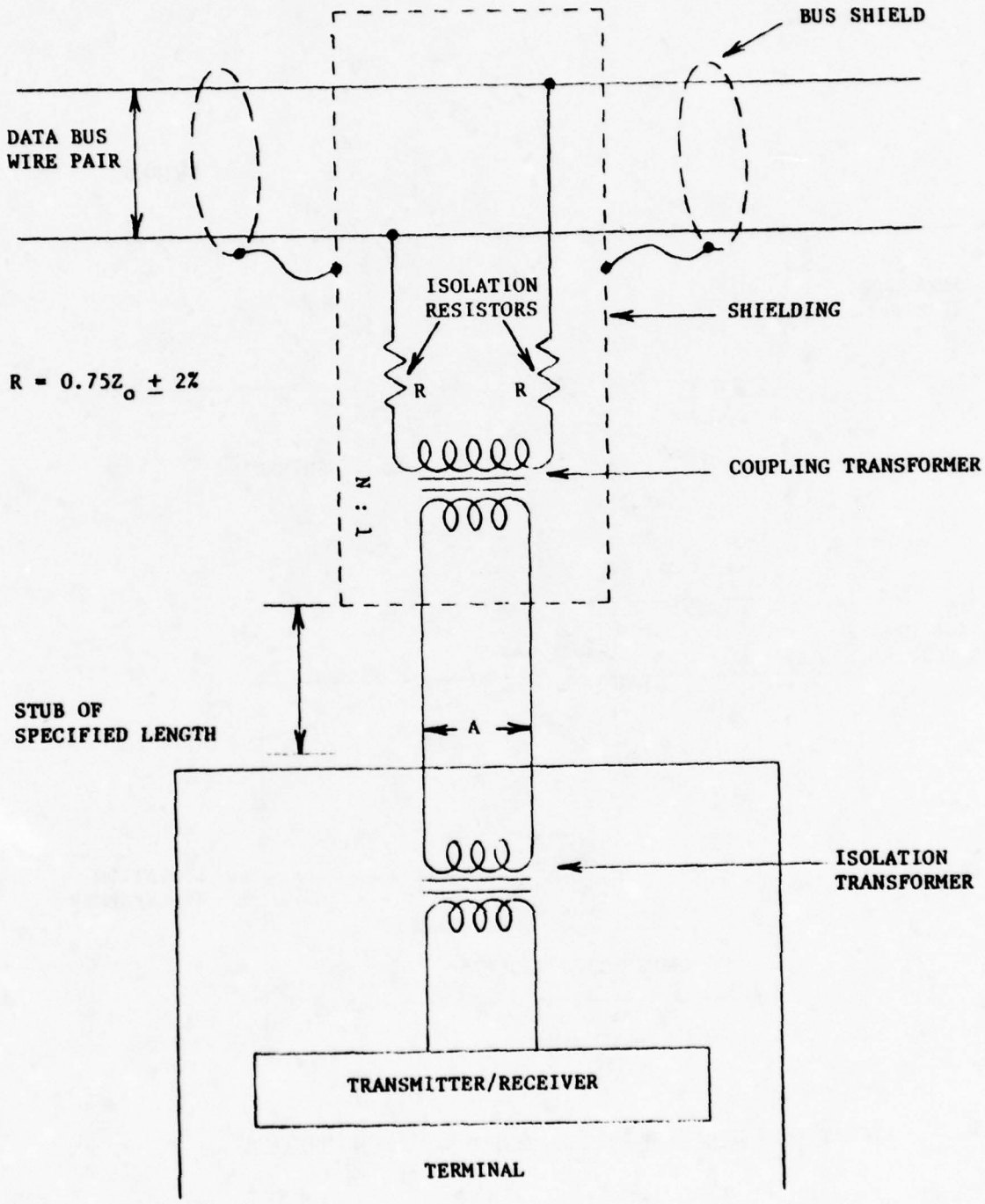



FIGURE 9. Data bus interface using transformer coupling.

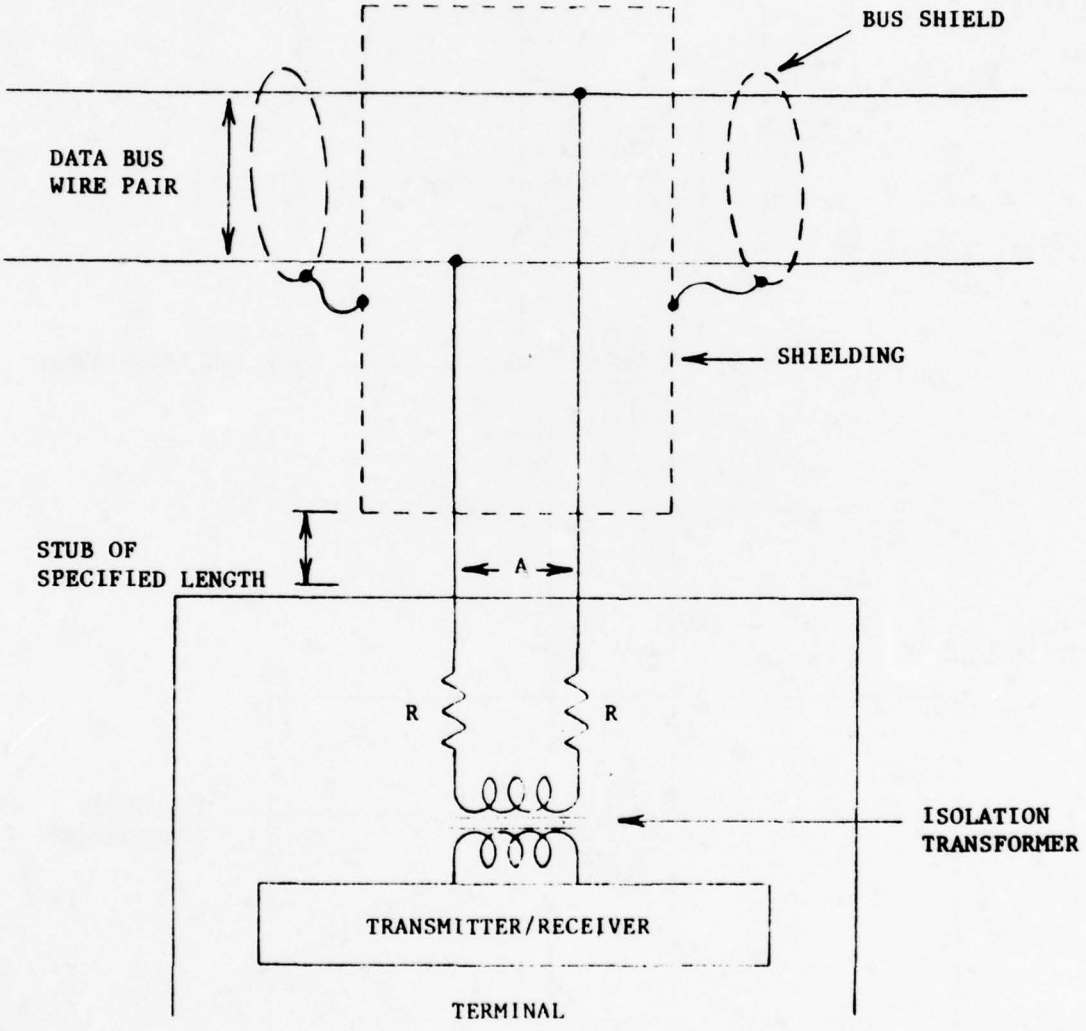



FIGURE 10. Data bus interface using direct coupling.

4.4.1.1 Word validation. The terminal shall insure that each word conforms to the following minimum criteria:

a. The word begins with a valid sync field.
b. The bits are in a valid Manchester II code.
c. The information field has 16 bits plus parity.
d. The word parity is odd.

When a word fails to conform to the preceding criteria, the word shall be considered invalid.

4.4.1.2 Transmission continuity. The terminal shall verify that the message is contiguous as defined in 4.3.3.6. Improperly timed data syncs shall be considered a message error.

4.4.1.3 Terminal fail-safe. The terminal shall contain a hardware implemented time-out to preclude a signal transmission of greater than 800.0 μs. This hardware shall not preclude a correct transmission in response to a command. Reset of this time-out function shall be performed by the reception of a valid command on the bus on which the time-out has occurred.

4.4.2 Bus controller operation. A terminal operating as a bus controller shall be responsible for sending data bus commands, participating in data transfers, receiving status responses, and monitoring system status as defined in this standard. the bus controller function may be embodied as either a stand-alone terminal, whose sole function is to control the data bus(s), or contained within a subsystem. Only one terminal shall be in active control of a data bus at any one time.

4.4.3 Remote terminal.

4.4.3.1 Operation. A remote terminal (RT) shall operate in response to valid commands received from the bus controller. The RT shall accept a command word as valid when the command word meets the criteria of 4.4.1.1, and the command word contains a terminal address which matches the RT address or an address of 11111, if the RT has the broadcast option.

4.4.3.2 Superseding valid commands. The RT shall be capable of receiving a command word on the data bus after the minimum intermessage gap time as specified in 4.3.3.7 has been exceeded, when the RT is not in the time period T as specified in 4.3.3.8 prior to the transmission of a status word, and when it is not transmitting on that data bus. A second valid command word sent to an RT shall take precedence over the previous command. The RT shall respond to the second valid command as specified in 4.3.3.8.

4.4.3.3 Invalid commands. A remote terminal shall not respond to a command word which fails to meet the criteria specified in 4.4.3.1.

4.4.3.4 Illegal command. An illegal command is a valid command as specified in 4.4.3.1, where the bits in the subaddress/mode field, data word count/mode code field, and the T/R bit indicate a mode command, subaddress, or word count that has not been implemented in the RT. It is the responsibility of the bus controller to assure that no illegal commands are sent out. The RT designer has the option of monitoring for illegal commands. If an RT that is designed with this option detects an illegal command and the proper number of contiguous

valid data words as specified by the illegal command word, it shall respond with a status word only, setting the message error bit, and not use the information received.

4.4.3.5 Valid data reception. The remote terminal shall respond with a status word when a valid command word and the proper number of contiguous valid data words are received, or a single valid word associated with a mode code is received. Each data word shall meet the criteria specified in 4.4.1.1.

4.4.3.6 Invalid data reception. Any data word(s) associated with a valid receive command that does not meet the criteria specified in 4.4.1.1 and 4.4.1.2 or an error in the data word count shall cause the remote terminal to set the message error bit in the status word to a logic one and suppress the transmission of the status word. If a message error has occurred, then the entire message shall be considered invalid.

4.4.4 Bus monitor operation. A terminal operating as a bus monitor shall receive bus traffic and extract selected information. While operating as a bus monitor, the terminal shall not respond to any message except one containing its own unique address if one is assigned. All information obtained while acting as a bus monitor shall be strictly used for off-line applications (e.g., flight test recording, maintenance recording or mission analysis) or to provide the back-up bus controller sufficient information to take over as the bus controller.

4.5 Hardware characteristics.

4.5.1 Data bus characteristics.

4.5.1.1 Cable. The cable used for the main bus and all stubs shall be a two conductor, twisted, shielded, jacketed cable. The wire-to-wire distributed capacitance shall not exceed 30.0 picofarads per foot. The cables shall be formed with not less than four twists per foot where a twist is defined as a 360 degree rotation of the wire pairs; and, the cable shield shall provide a minimum of 75.0 percent coverage.

4.5.1.2 Characteristic impedance. The nominal characteristic impedance of the cable ($Z_o$) shall be within the range of 70.0 ohms to 85.0 ohms at a sinusoidal frequency of 1.0 megahertz (MHz).

4.5.1.3 Cable attenuation. At the frequency of 4.5.1.2, the cable power loss shall not exceed 1.5 decibels (dB)/100 feet (ft).

4.5.1.4 Cable termination. The two ends of the cable shall be terminated with a resistance, equal to the selected cable nominal characteristic impedance ($Z_o$) $\pm$ 2.0 percent.

4.5.1.5 Cable stub requirements. The cable shall be coupled to the terminal as shown on figure 9 or figure 10. The use of long stubs is discouraged, and the length of a stub should be minimized. However, if installation requirements dictate, stub lengths exceeding those lengths specified in 4.5.1.5.1 and 4.5.1.5.2 are permissible.

4.5.1.5.1 Transformer coupled stubs. The length of a transformer coupled stub should not exceed 20 feet. If a transformer coupled stub is used, then the following shall apply.

4.5.1.5.1.1 Coupling transformer. A coupling transformer, as shown on figure 9, shall be required. This transformer shall have a turns ratio of 1:1.41 ± 3.0 percent, with the higher turns on the isolation resistor side of the stub.

4.5.1.5.1.1.1 Transformer input impedance. The open circuit impedance as seen at point B on figure 11 shall be greater than 3000 ohms over the frequency range of 75.0 kilohertz (kHz) to 1.0 megahertz (MHz), when measured with a 1.0 V root-mean-square (RMS) sin wave.

4.5.1.5.1.1.2 Transformer waveform integrity. The droop of the transformer using the test configuration shown on figure 11 at point B, shall not exceed 20.0 percent. Overshoot and ringing as measured at point B shall be less that ± 1.0 V peak. For this test, R shall equal 360.0 ohms ± 5.0 percent and the input A of figure 11 shall be a 250.0 kHz square wave. 27.0 V peak-to-peak, with a rise and fall time no greater than 100 nanoseconds (ns).

4.5.1.5.1.1.3 Transformer common mode rejection. The coupling transformer shall have a common mode rejection ratio greater than 45.0 dB at 1.0 MHz.

4.5.1.5.1.2 Fault isolation. An isolation resistor shall be placed in series with each connection to the data bus cable. This resistor shall have a value of 0.75 $Z_o$ ohms plus or minus 2.0 percent, where $Z_o$ is the selected cable nominal characteristic impedance. The impedance placed across the data bus cable shall be no less than 1.5 $Z_o$ ohms for any failure of the coupling transformer, cable stub, or terminal transmitter/receiver.

4.5.1.5.1.3 Cable coupling. All coupling transformers and isolation resistors, as specified in 4.5.1.5.1.1 and 4.5.1.5.1.2, shall have continuous shielding which will provide a minimum of 75 percent coverage. The isolation resistors and coupling transformers shall be placed at minimum possible distance from the junction of the stub to the main bus.

4.5.1.5.1.4 Stub voltage requirements. Every data bus shall be designed such that all stubs at point A of figure 9 shall have a peak-to-peak amplitude, line-to-line within the range of 1.0 and 14.0 V for a transmission by any terminal on the data bus. This shall include the maximum reduction of data bus signal amplitude in the event that one of the terminals has a fault which causes it to reflect a fault impedance specified in 4.5.1.5.1.2 on the data bus. This shall also include the worse case output voltage of the terminals as specified in 4.5.2.1.1.1 and 4.5.2.2.1.1.

4.5.1.5.2 Direct coupled stubs. The length of a direct coupled stub should not exceed 1 foot. Refer to 10.5 for comments concerning direct coupled stubs. If a direct coupled stub is used, then the following shall apply.

4.5.1.5.2.1 Fault isolation. An isolation resistor shall be placed in series with each connection to the data bus cable. This resistor shall have a value of 55.0 ohms plus or minus 2.0 percent. The isolation resistors shall be placed within the RT as shown on figure 10.

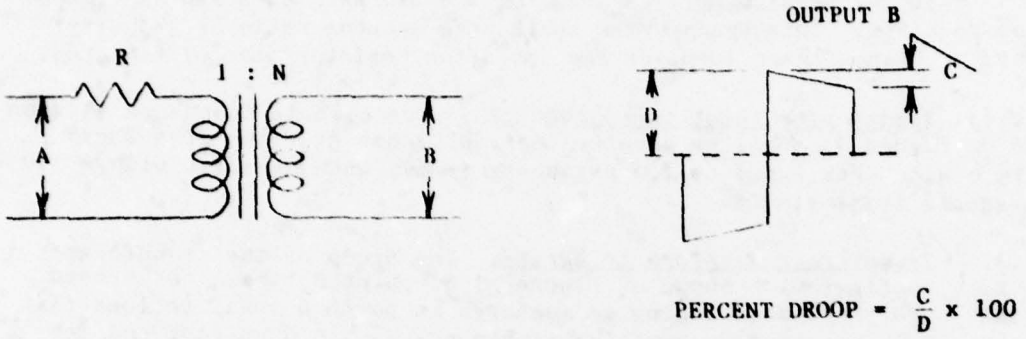



FIGURE 11. Coupling transformer.

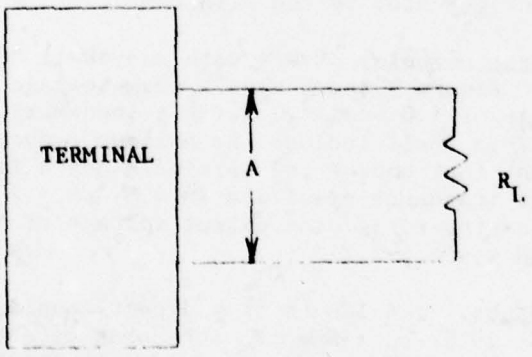



FIGURE 12. Terminal I/O characteristics for transformer coupled and direct coupled stubs.

4.5.1.5.2.2 Cable coupling. All bus-stub junctions shall have continuous shielding which will provide a minimum of 75 percent coverage.

4.5.1.5.2.3 Stub voltage requirements. Every data bus shall be designed such that all stubs at point A of figure 10 shall have a peak-to-peak amplitude, line-to-line within the range of 1.4 and 20.0 V for a transmission by any terminal on the data bus. This shall include the maximum reduction of data bus signal amplitude in the event that one of the terminals has a fault which causes it to reflect a fault impedance of 110 ohms on the data bus. This shall also include the worst case output voltage of the terminals as specified in 4.5.2.1.1.1 and 4.5.2.2.1.1.

4.5.1.5.3 Wiring and cabling for EMC. For purposes of electromagnetic capability (EMC), the wiring and cabling provisions of MIL-E-6051 shall apply.

4.5.2 Terminal characteristics.

4.5.2.1 Terminals with transformer coupled stubs.

4.5.2.1.1 Terminal output characteristics. The following characteristics shall be measured with $R_L$, as shown on figure 12, equal to 70.0 ohms ± 2.0 percent.

4.5.2.1.1.1 Output levels. The terminal output voltage levels shall be measured using the test configuration shown on figure 12. The terminal output voltage shall be within the range of 18.0 to 27.0 V, peak-to-peak, line-to-line, when measured at point A on figure 12.

4.5.2.1.1.2 Output waveform. The waveform, when measured at point A on figure 12 shall have zero crossing deviations which are equal to, or less than, 25.0 ns from the ideal crossing point, measured with respect to the previous zero crossing (i.e., .5 ± .025 µs, 1.0 ± .025 µs, 1.5 ± .025 µs, and 2.0 ± .025 µs). The rise and fall time of this waveform shall be from 100.0 to 300.0 ns when measured from levels of 10 to 90 percent of full waveform peak-to-peak, line-to-line, voltage as shown on figure 13. Any distortion of the waveform including overshoot and ringing shall not exceed ± 900.0 millivolts (mV) peak, line-to-line, as measured at point A, figure 12.

4.5.2.1.1.3 Output noise. Any noise transmitted when the terminal is receiving or has power removed, shall not exceed a value of 14.0 mV, RMS, line-to-line, as measured at point A, figure 12.

4.5.2.1.1.4 Output symmetry. From the time beginning 2.5 us after the mid-bit crossing of the parity bit of the last word transmitted by a terminal, the maximum voltage at point A of figure 12 shall be no greater than ± 250.0 mV peak, line-to-line. This shall be tested with the terminal transmitting the maximum number of words it is designed to transmit, up to 33. This test shall be run six times with each word in a contiguous block of words having the same bit pattern. The six word contents that shall be used are $8000_{16}$, $7FFF_{16}$, $0000_{16}$, $FFFF_{16}$, $5555_{16}$, and $AAAA_{16}$. The output of the terminal shall be as specified in 4.5.2.1.1.1 and 4.5.2.1.1.2.

4.5.2.1.2 Terminal input characteristics. The following characteristics shall be measured independently.

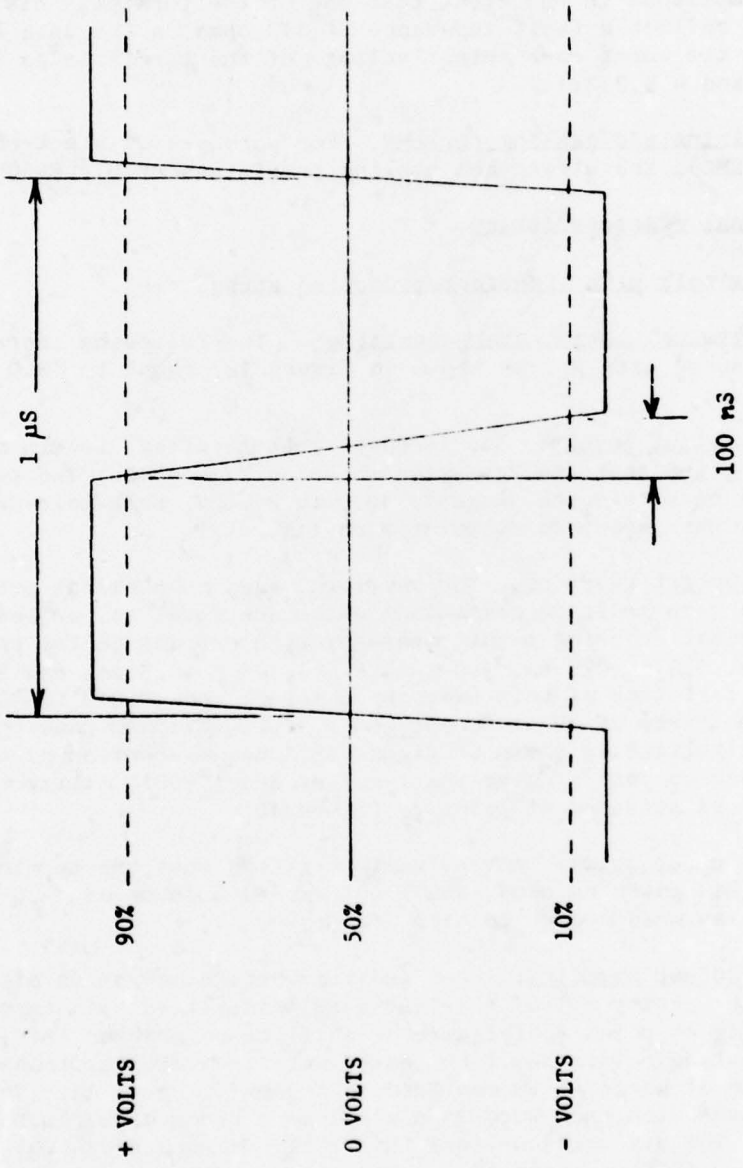



FIGURE 13. Output waveform.

4.5.2.1.2.1 Input waveform compatibility. The terminal shall be capable of receiving and operating with the incoming signals specified herein, and shall accept waveform varying from a square wave to a sine wave with a maximum zero crossing deviation from the ideal with respect to the previous zero crossing of ± 150 ns, (i.e., 2.0 ± .15 µs, 1.5 ± .15 µs, 1.0 ± .15 µs, .5 ± .15 µs). The terminal shall respond to an input signal whose peak-to-peak amplitude, line-to-line, is within the range of .86 to 14.0 V. The terminal shall not respond to an input signal whose peak-to-peak amplitude, line-to-line, is within the range of 0.0 to .20 V. The voltages are measured at point A on figure 9.

4.5.2.1.2.2 Common mode rejections. Any signals from direct current (DC) to 2.0 MHz, with amplitudes equal to or less than ± 10.0 V peak, line-to-ground, measured at point A on figure 9, shall not degrade the performance of the receiver.

4.5.2.1.2.3 Input impedance. The magnitude of the terminal input impedance, when the RT is not transmitting, or has power removed, shall be a minimum of 1000.0 ohms within the frequency range of 75.0 khz to 1.0 MHz. This impedance is that measured line-to-line at point A on figure 9.

4.5.2.1.2.4 Noise rejection. The terminal shall exhibit a maximum word error rate of one part in $10^7$, on all words received by the terminal, after validation checks as specified in 4.4, when operating in the presence of additive white Gaussian noise distributed over a bandwidth of 1.0 kHz to 4.0 MHz at an RMS amplitude of 140 mV. A word error shall include any fault which causes the message error bit to be set in the terminal's status word, or one which causes a terminal to not respond to a valid command. The word error rate shall be measured with a 2.1 V peak-to-peak, line-to-line, input to the terminal as measured at point A on figure 9. The noise tests shall be run continuously until, for a particular number of failures, the number of words received by the terminal, including both command and data words, exceeds the required number for acceptance of the terminal, or is less than the required number for rejection of the terminal, as specified in table II. All data words used in the tests shall contain random bit patterns. These bit patterns shall be unique for each data word in a message, and shall change randomly from message to message.

4.5.2.2 Terminals with direct coupled stubs.

4.5.2.2.1 Terminal output characteristics. The following characteristics shall be measured with $R_L$, as shown on figure 12, equal to 35.0 ohms ± 2.0 percent.

4.5.2.2.1.1 Output levels. The terminal output voltage levels shall be measured using the test configuration shown on figure 12. The terminal output voltage shall be within the range of 6.0 to 9.0 V, peak-to-peak, line-to-line, when measured at point A on figure 12.

Table II. Criteria for acceptance or rejection of a terminal for the noise rejection test

| No of Errors | TOTAL WORDS RECEIVED BY THE TERMINAL (in multiples of $10^7$) Reject (Equal or less) | Accept (Equal or more) |
|---|---|---|
| 0 | N/A | 4.40 |
| 1 | N/A | 5.21 |
| 2 | N/A | 6.02 |
| 3 | N/A | 6.83 |
| 4 | N/A | 7.64 |
| 5 | N/A | 8.45 |
| 6 | .45 | 9.27 |
| 7 | 1.26 | 10.08 |
| 8 | 2.07 | 10.89 |
| 9 | 2.88 | 11.70 |
| 10 | 3.69 | 12.51 |
| 11 | 4.50 | 13.32 |
| 12 | 5.31 | 14.13 |
| 13 | 6.12 | 14.94 |
| 14 | 6.93 | 15.75 |
| 15 | 7.74 | 16.56 |
| 16 | 8.55 | 17.37 |
| 17 | 9.37 | 18.19 |
| 18 | 10.18 | 19.00 |
| 19 | 10.99 | 19.81 |
| 20 | 11.80 | 20.62 |
| 21 | 12.61 | 21.43 |
| 22 | 13.42 | 22.24 |
| 23 | 14.23 | 23.05 |
| 24 | 15.04 | 23.86 |
| 25 | 15.85 | 24.67 |
| 26 | 16.66 | 25.48 |
| 27 | 17.47 | 26.29 |
| 28 | 18.29 | 27.11 |
| 29 | 19.10 | 27.92 |
| 30 | 19.90 | 28.73 |
| 31 | 20.72 | 29.54 |
| 32 | 21.53 | 30.35 |
| 33 | 22.34 | 31.16 |
| 34 | 23.15 | 31.97 |
| 35 | 23.96 | 32.78 |
| 36 | 24.77 | 33.00 |
| 37 | 25.58 | 33.00 |
| 38 | 26.39 | 33.00 |
| 39 | 27.21 | 33.00 |
| 40 | 28.02 | 33.00 |
| 41 | 33.00 | N/A |

4.5.2.2.1.2 Output waveform. The waveform, when measured at point A on figure 12, shall have zero crossing deviations which are equal to, or less than, 25.0 ns from the ideal crossing point, measured with respect to the previous zero crossing (i.e., .5 ± .025 µs, 1.0 ± .025 µs, 1.5 ± .025 µs and 2.0 ± .025 µs). The rise and fall time of this waveform shall be from 100.0 to 300.0 ns when measured from levels of 10 to 90 percent of full waveform peak-to-peak, line-to-line, voltage as shown on figure 13. Any distortion of the waveform including overshoot and ringing shall not exceed ± 300.0 mV peak, line-to-line, as measured at point A on figure 12.

4.5.2.2.1.3 Output noise. Any noise transmitted when the terminal is receiving or has power removed, shall not exceed a value of 5.0 mV, RMS, line-to-line, as measured at point A on figure 12.

4.5.2.2.1.4 Output symmetry. From the time beginning 2.5 us after the mid-bit crossing of the parity bit of the last word transmitted by a terminal, the maximum voltage at point A on figure 12, shall be no greater than ± 90.0 mV peak, line-to-line. This shall be tested with the terminal transmitting the maximum number of words it is designed to transmit, up to 33. This test shall be run six times with each word in a contiguous block of words having the same bit pattern. The six word contents that shall be used are $8000_{16}$, $7FFF_{16}$, $0000_{16}$, $FFFF_{16}$, $5555_{16}$, and $AAAA_{16}$. The output of the terminal shall be as specified in 4.5.2.2.1.1 and 4.5.2.2.1.2.

4.5.2.2.2 Terminal input characteristics. The following characteristics shall be measured independently.

4.5.2.2.2.1 Input waveform compatibility. The terminal shall be capable of receiving and operating with the incoming signals specified herein, and shall accept waveform varying from a square wave to a sine wave with a maximum zero crossing deviation from the ideal with respect to the previous zero crossing of plus or minus 150 ns, (i.e., 2.0 ± .15 us 1.5 ± .15 us 1.0 ± .15 us .5 ± .15 us). The terminal shall respond to an input signal whose peak-to-peak amplitude, line-to-line, is within the range of 1.2 to 20.0 V. The terminal shall not respond to an input signal whose peak-to-peak amplitude, line-to-line, is within the range of 0.0 to .28 V. The voltages are measured at point A on figure 10.

4.5.2.2.2.2 Common mode rejections. Any signals from DC to 2.0 MHz, with amplitudes equal to or less than ± 10.0 V peak, line-to-ground, measured at point A on figure 10, shall not degrade the performance of the receiver.

4.5.2.2.2.3 Input impedance. The magnitude of the terminal input impedance, when the RT is not transmitting, or has power removed, shall be a minimum of 2000.0 ohms within the frequency range of 75.0 kHz to 1.0 MHz. This impedance is that measured line-to-line at point A on figure 10.

4.5.2.2.2.4 Noise rejection. The terminal shall exhibit a maximum word error rate of one part in $10^7$, on all words received by the terminal, after validation checks as specified in 4.4, when operating in the presence of additive white Gaussian noise distributed over a bandwidth of 1.0 kHz to 4.0 MHz at an RMS amplitude of 200 mV. A word error shall include any fault which causes the message error bit to be set in the terminal's status word, or one which causes a terminal to not respond to a valid command. The word error rate shall be measured with a 3.0 V peak-to-peak, line-to-line, input to the

terminal as measured at point A on figure 10. The noise tests shall be run continuously until, for a particular number of failures, the number of words received by the terminal, including both command and data words, exceeds the required number for acceptance of the terminal, or is less than the required number for rejection of the terminal, as specified in table II. All data words used in the tests shall contain random bit patterns. These bit patterns shall be unique for each data word in a message, and shall change randomly from message to message.

4.6 Redundant data bus requirements. If redundant data buses are used, the requirements as specified in the following shall apply to those data buses.

4.6.1 Electrical isolation. All terminals shall have a minimum of 45 dB isolation between data buses. Isolation here means the ratio in dB between the output voltage on the active data bus and the output voltage on the inactive data bus. This shall be measured using the test configuration specified in 4.5.2.1.1 or 4.5.2.2.1 for each data bus. Each data bus shall be alternately activated with all measurements being taken at point A on figure 12 for each data bus.

4.6.2 Single event failures. All data buses shall be routed to minimize the possibility that a single event failure to a data bus shall cause the loss of more than that particular data bus.

4.6.3 Dual standby redundant data bus. If a dual redundant data bus is used, then it shall be a dual standby redundant data bus as specified in the following paragraphs.

4.6.3.1 Data bus activity. Only one data bus can be active at any given time except as specified in 4.6.3.2.

4.6.3.2 Reset data bus transmitter. If while operating on a command, a terminal receives another valid command, from either data bus, it shall reset and respond to the new command on the data bus on which the new command is received. The terminal shall respond to the new command as specified in 4.3.3.8.

5. DETAIL REQUIREMENTS (Not Applicable)

Custodians:
Army - EL
Navy - AS
Air Force - 11

Preparing Activity:
Air Force - 11

Project MISC-0D03

# APPENDIX

10. General. The following paragraphs in this appendix are presented in order to discuss certain aspects of the standard in a general sense. They are intended to provide a user of the standard more insight into the aspects discussed.

10.1 Redundancy. It is intended that this standard be used to support rather than to supplant the system design process. However, it has been found, through application experience in various aircraft, that the use of a dual standby redundancy technique is very desirable for use in integrating mission avionics. For this reason, this redundancy scheme is defined in 4.6 of this standard. None the less, the system designer should utilize this standard as the needs of a particular application dictate. The use of redundancy, the degree to which it is implemented, and the form which it takes must be determined on an individual application basis. Figures 10.1 and 10.2 illustrate some possible approaches to dual redundancy. These illustrations are not intended to be inclusive, but rather representative. It should be noted that analogous approaches exist for the triple and quad redundant cases.

10.2 Bus controller. The bus controller is a key part of the data bus system. The functions of the bus controller, in addition to the issuance of commands, must include the constant monitoring of the data bus and the traffic on the bus. It is envisioned that most of the routine minute details of bus monitoring (e.g., parity checking, terminal non-response time-out, etc.) will be embodied in hardware, while the algorithms for bus control and decision making will reside in software. It is also envisioned that, in general, the bus controller will be a general purpose airborne computer with a special input/output (I/O) to interface with the data bus. It is of extreme importance in bus controller design that the bus controller be readily able to accommodate terminals of differing protocol's and status word bits used. Equipment designed to MIL-STD-1553A will be in use for a considerable period of time; thus, bus controllers must be capable of adjusting to their differing needs. It is also important to remember that the bus controller will be the focal point for modification and growth within the multiplex system, and thus the software must be written in such a manner as to permit modification with relative ease.

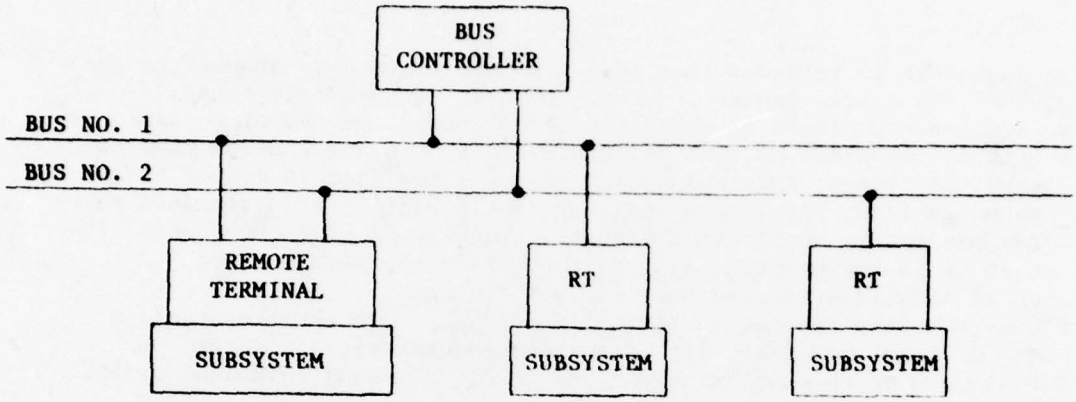



FIGURE 10.1. Illustration of possible redundancy.

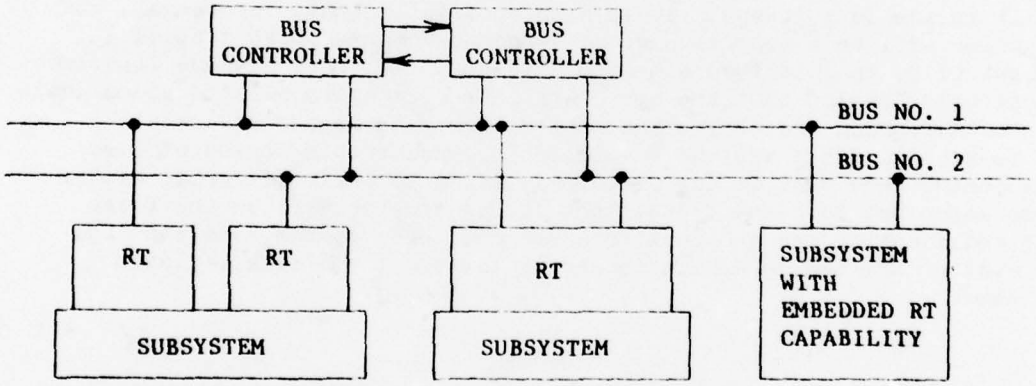



Figure 10.2

NOTE: RT - Remote Terminal

FIGURE 10.2. Illustration of possible redundancy.

10.3 Multiplex selection criteria. The selection of candidate signals for multiplexing is a function of the particular application involved, and criteria will in general vary from system to system. Obviously, those signals which have bandwidths of 400 Hz or less are prime candidates for inclusion on the bus. It is also obvious that video, audio, and high speed parallel digital signals should be excluded. The area of questionable application is usually between 400 Hz and 3KHz bandwidth. The transfer of these signals on the data bus will depend heavily upon the loading of the bus in a particular application. The decision must be based on projected future bus needs as well as the current loading. Another class of signals which in general are not suitable for multiplexing are those which can be typified by a low rate (over a mission) but possessing a high priority or urgency. Examples of such signals might be a nuclear event detector output or a missile launch alarm from a warning receiver. Such signals are usually better left hardwired, but they may be accommodated by the multiplex system if a direct connection to the bus controller's interrupt hardware is used to trigger a software action in response to the signal.

10.4 High reliability requirements. The use of simple parity for error detection within the multiplex bus system was dictated by a compromise between the need for reliable data transmission, system overhead, and remote terminal simplicity. Theoretical and empirical evidence indicates that an undetected bit error rate of $10^{-12}$ can be expected from a practical multiplex system built to this standard. If a particular signal requires a bit error rate which is better than that provided by the parity checking, then it is incumbent upon the system designer to provide the reliability within the constraints of the standard or to not include this signal within the multiplex bus system. A possible approach in this case would be to have the signal source and sink provide appropriate error detection and correction encoding/decoding and employ extra data words to transfer the information. Another approach would be to partition the message, transmit a portion at a time, and then verify (by interrogation) the proper transfer of each segment.

10.5 Stubbing. Stubbing is the method wherein a separate line is connected between the primary data bus line and a terminal. The direct connection of a stub line causes a mismatch which appears on the waveforms. This mismatch can be reduced by filtering at the receiver and by using bi-phase modulation. Stubs are often employed not only as a convenience in bus layout but as a means of coupling a unit to the line in such a manner that a fault on the stub or terminal will not greatly affect the transmission line operation. In this case, a network is employed in the stub line to provide isolation from the fault. These networks are also used for stubs that are of such length that the mismatch and reflection degrades bus operation. The preferred method of stubbing is to use transformer coupled stubs, as defined in 4.5.1.5.1. This method provides the benefits of DC isolation, increased common mode protection, a doubling of effective stub impedance, and fault isolation for the entire stub and terminal. Direct coupled stubs, as defined in 4.5.1.5.2 of this standard, should be avoided if at all possible. Direct coupled stubs provide no DC isolation or common mode rejection for the terminal external to its subsystem. Further, any shorting fault between the subsystems internal isolation resistors (usually on a circuit board) and the main bus junction will cause failure of that entire bus. It can be expected that when the direct coupled stub length exceeds 1.6 feet, that it will begin to distort the main bus waveforms. Note that this length includes the cable runs internal to a given subsystem.

10.6 Use of broadcast option. The use of a broadcast message as defined in 4.3.3.6.7 of this standard represents a significant departure from the basic philosophy of this standard in that it is a message format which does not provide positive closed-loop control of bus traffic. The system designer is strongly encouraged to solve any design problems through the use of the three basic message formats without resorting to use of the broadcast. If system designers do choose to use the broadcast command, they should carefully consider the potential effects of a missed broadcast message, and the subsequent implications for fault or error recovery design in the remote terminals and bus controllers.



*U.S.Government Printing Office: 1979 — 657-002/484

## STANDARDIZATION DOCUMENT IMPROVEMENT PROPOSAL

OMB Approval
No. 22-R255

INSTRUCTIONS: The purpose of this form is to solicit beneficial comments which will help achieve procurement of suitable products at reasonable cost and minimum delay, or will otherwise enhance use of the document. DoD contractors, government activities, or manufacturers/vendors who are prospective suppliers of the product are invited to submit comments to the government. Fold on lines on reverse side, staple in corner, and send to preparing activity. Comments submitted on this form do not constitute or imply authorization to waive any portion of the referenced document(s) or to amend contractual requirements. Attach any pertinent data which may be of use in improving this document. If there are additional papers, attach to form and place both in an envelope addressed to preparing activity.

DOCUMENT IDENTIFIER AND TITLE: MIL-STD-1553B

NAME OF ORGANIZATION AND ADDRESS

CONTRACT NUMBER

MATERIAL PROCURED UNDER A

☐ DIRECT GOVERNMENT CONTRACT ☐ SUBCONTRACT

1. HAS ANY PART OF THE DOCUMENT CREATED PROBLEMS OR REQUIRED INTERPRETATION IN PROCUREMENT USE?

A. GIVE PARAGRAPH NUMBER AND WORDING

B. RECOMMENDATIONS FOR CORRECTING THE DEFICIENCIES

2. COMMENTS ON ANY DOCUMENT REQUIREMENT CONSIDERED TOO RIGID

3. IS THE DOCUMENT RESTRICTIVE?

☐ YES ☐ NO *(If "Yes", in what way?)*

4. REMARKS

SUBMITTED BY *(Printed or typed name and address - Optional)*

TELEPHONE NO.

DATE

DD FORM 1 JAN 72 1426

REPLACES EDITION OF 1 JAN 66 WHICH MAY BE USED

C16061

FOLD

ASD/ENESS
Wright-Patterson AFB, O 45433
OFFICIAL BUSINESS
PENALTY FOR PRIVATE USE $300

POSTAGE AND FEES PAID
DEPARTMENT OF THE AIR FORCE
DoD-318



ASD/ENESS
Wright-Patterson AFB, Ohio 45433

FOLD